

|                |                |              |                |            |              |
|----------------|----------------|--------------|----------------|------------|--------------|
| DDDDDDDDDDDDDD | EEEEEEEEEFFFFE | BBBBBBBBBBBB | UUU            | UUU        | GGGGGGGGGGGG |
| DDDDDDDDDDDDDD | EEEEEEEEEFFFFE | BBBBBBBBBBBB | UUU            | UUU        | GGGGGGGGGGGG |
| DDDDDDDDDDDDDD | EEEEEEEEEFFFFE | BBBBBBBBBBBB | UUU            | UUU        | GGGGGGGGGGGG |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDD            | DDD EEE        | BBB          | BBB            | UUU        | UUU GGG      |
| DDDDDDDDDDDDDD | EEEEEEEEEFFFFE | BBBBBBBBBBBB | UUUUUUUUUUUUUU | GGGGGGGGGG |              |
| DDDDDDDDDDDDDD | EEEEEEEEEFFFFE | BBBBBBBBBBBB | UUUUUUUUUUUUUU | GGGGGGGGGG |              |
| DDDDDDDDDDDDDD | EEEEEEEEEFFFFE | BBBBBBBBBBBB | UUUUUUUUUUUUUU | GGGGGGGGGG |              |

|          |          |          |        |    |          |    |    |           |           |
|----------|----------|----------|--------|----|----------|----|----|-----------|-----------|
| DDDDDDDD | BBBBBBBB | GGGGGGGG | NN     | NN | TTTTTTTT | YY | YY | PPPPPPP   | EEEEEEEEE |
| DDDDDDDD | BBBBBBBB | GGGGGGGG | NN     | NN | TTTTTTTT | YY | YY | PPPPPPP   | EEEEEEEEE |
| DD DD    | BB BB    | GG       | NN     | NN | TT       | YY | YY | PP        | EE        |
| DD DD    | BB BB    | GG       | NN     | NN | TT       | YY | YY | PP        | EE        |
| DD DD    | BB BB    | GG       | NNNN   | NN | TT       | YY | YY | PP        | EE        |
| DD DD    | BB BB    | GG       | NNNN   | NN | TT       | YY | YY | PP        | EE        |
| DD DD    | BBBBBBBB | GG       | NN NN  | NN | TT       | YY | YY | PPPPPPP   | EEEEEEEEE |
| DD DD    | BBBBBBBB | GG       | NN NN  | NN | TT       | YY | YY | PPPPPPP   | EEEEEEEEE |
| DD DD    | BB BB    | GG       | GGGGGG | NN | NNNN     | TT | YY | PP        | EE        |
| DD DD    | BB BB    | GG       | GGGGGG | NN | NNNN     | TT | YY | PP        | EE        |
| DD DD    | BB BB    | GG       | GG     | NN | NN       | TT | YY | PP        | EE        |
| DD DD    | BB BB    | GG       | GG     | NN | NN       | TT | YY | PP        | EE        |
| DDDDDDDD | BBBBBBBB | GGGGGG   | NN     | NN | TT       | YY | PP | EEEEEEEEE | ....      |
| DDDDDDDD | BBBBBBBB | GGGGGG   | NN     | NN | TT       | YY | PP | EEEEEEEEE | ....      |

|          |        |          |
|----------|--------|----------|
| LL       | IIIIII | SSSSSSSS |
| LL       | IIIIII | SSSSSSSS |
| LL       | II     | SS       |
| LL       | II     | SS       |
| LL       | II     | SS       |
| LL       | II     | SSSSSS   |
| LL       | II     | SSSSSS   |
| LL       | II     | SS       |
| LL       | II     | SS       |
| LL       | II     | SS       |
| LLLLLLLL | IIIIII | SSSSSSSS |
| LLLLLLLL | IIIIII | SSSSSSSS |

1 0001 0 MODULE DBGNTYPE (IDENT = 'V04-000') =  
2 0002 0  
3 0003 1 BEGIN  
4 0004 1  
5 0005 1  
6 0006 1 \*\*\*\*\*  
7 0007 1 \*  
8 0008 1 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
9 0009 1 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
10 0010 1 \* ALL RIGHTS RESERVED.  
11 0011 1 \*  
12 0012 1 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
13 0013 1 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
14 0014 1 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
15 0015 1 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
16 0016 1 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
17 0017 1 \* TRANSFERRED.  
18 0018 1 \*  
19 0019 1 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
20 0020 1 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
21 0021 1 \* CORPORATION.  
22 0022 1 \*  
23 0023 1 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
24 0024 1 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
25 0025 1 \*  
26 0026 1 \*  
27 0027 1 \*\*\*\*\*  
28 0028 1 :  
29 0029 1 :  
30 0030 1 :++  
31 0031 1 :  
32 0032 1 :  
33 0033 1 :  
34 0034 1 :  
35 0035 1 :  
36 0036 1 :  
37 0037 1 :  
38 0038 1 :  
39 0039 1 :  
40 0040 1 :  
41 0041 1 :  
42 0042 1 :  
43 0043 1 :  
44 0044 1 :  
45 0045 1 :  
46 0046 1 :  
47 0047 1 :  
48 0048 1 :  
49 0049 1 :  
50 0050 1 :  
51 0051 1 :  
52 0052 1 :  
53 0053 1 :  
54 0054 1 :  
55 0055 1 :  
56 0056 1 :  
57 0057 1 :  
FACILITY:  
DEBUG  
ABSTRACT:  
This module contains the parse and execution networks for the TYPE  
verb. The parsing method employed is that of ATN's.  
ENVIRONMENT:  
VAX/VMS  
AUTHOR:  
Richard Title  
CREATION DATE:  
9-14-81  
VERSION:  
V03.0-001  
MODIFIED BY:  
V. Wn't May, 1982

58 0058 1 |  
59 0059 1 | REVISION HISTORY:  
60 0060 1 | 3.10 14-May-82 VJH Added call to DBG\$FLUSHBUF, eliminating need to  
61 0061 1 | initialize local output buffer.  
62 0062 1 | 3.11 2-Jun-82 VJH Removed all references to DBG\$FAO\_PUT and  
63 0063 1 | DBG\$OUT\_PUT, replacing them with calls to  
64 0064 1 | DBG\$PRINT and DBG\$NEWLINE, respectively.  
65 0065 1 | --

```
67 0066 1 TABLE OF CONTENTS:  
68 0067 1  
69 0068 1  
70 0069 1  
71 0070 1 FORWARD ROUTINE  
72 0071 1 DBGSNPARSE_TYPE,  
73 0072 1 DBGSNEXECUTE_TYPE,  
74 0073 1 DBGSNPARSE_EDIT,  
75 0074 1 DBGSNEXECUTE_EDIT;  
76 0075 1  
77 0076 1  
78 0077 1 REQUIRE FILES:  
79 0078 1  
80 0079 1  
81 0080 1 REQUIRE 'SRC$:DBGPROLOG.REQ';  
82 0214 1  
83 0215 1 LIBRARY 'LIB$:DBGGEN.L32';  
84 0216 1  
85 0217 1 EXTERNAL ROUTINE  
86 0218 1 dbg$sta_getsourcmod,  
87 0219 1 dbg$sta_symname,  
88 0220 1  
89 0221 1 dbg$print: NOVALUE,  
90 0222 1 dbg$flushbuf: NOVALUE,  
91 0223 1 dbg$next_word,  
92 0224 1 dbg$newline: NOVALUE,  
93 0225 1 dbg$nmatch,  
94 0226 1 dbg$get_tempmem,  
95 0227 1 dbg$make_arg_vect,  
96 0228 1 dbg$nsave_decimal_integer,  
97 0229 1  
98 0230 1 dbg$nsyntax_error,  
99 0231 1  
100 0232 1 dbg$scr_screen_normal: NOVALUE,  
101 0233 1 dbg$src_line_to_rec,  
102 0234 1 dbg$src_type_lnum_source,  
103 0235 1  
104 0236 1  
105 0237 1 lib$do_command,  
106 0238 1 lib$spawn,  
107 0239 1 smg$set_keypad_mode,  
108 0240 1 sys$fao;  
109 0241 1  
110 0242 1 EXTERNAL  
111 0243 1 DBGSGB_KEYPAD INPUT: BYTE,  
112 0244 1 DBGSGL_KEYBOARD_ID,  
113 0245 1 DBGSGL_CONTROL: DBGSCONTROL_FLAGS,  
114 0246 1 DBGSGL_DEVELOPER: BITVECTOR[],  
115 0247 1 DBGSGL_SCREEN_SOURCE,  
116 0248 1  
117 0249 1 DBGSRC_NEXT_MODRSTPTR,  
118 0250 1  
119 0251 1 DBGSRC_NEXT_LNUM;  
120 0252 1  
121 0253 1  
122 0254 1  
123 0255 1 : These values can go in the VERB_COMPOSITE field of the EDIT verb  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255
```

DBGNTYPE  
V04-000

F 11  
16-Sep-1984 02:08:23  
14-Sep-1984 12:17:25  
VAX-11 Bliss-32 V4.0-742  
[DEBUG.SRC]DBGNTYPE.B32;1

Page 4  
(2)

: 124  
: 125  
: 126  
: 127  
: 128  
0256 1 !  
0257 1 LITERAL  
0258 1 EDIT\_EXIT = 1; | EDIT/EXIT  
0259 1 EDIT\_NOEXIT = 2; | EDIT/NOEXIT  
0260 1

```
130 0261 1 GLOBAL ROUTINE DBGSNPARSE_TYPE(INPUT_DESC, VERB_NODE, MESSAGE_VECT) =  
131 0262 1  
132 0263 1 Functional Description  
133 0264 1  
134 0265 1 ATN parse network for the TYPE verb.  
135 0266 1 This routine takes a verb node for the TYPE verb, and a string  
136 0267 1 descriptor for the remaining (unparsed) input.  
137 0268 1 It constructs a linked list of noun nodes, each of which  
138 0269 1 contains a module rst pointer and a line number range.  
139 0270 1  
140 0271 1 Formal Parameters  
141 0272 1  
142 0273 1 INPUT_DESC - A longword containing the address of the  
143 0274 1 command input descriptor.  
144 0275 1  
145 0276 1 VERB_NODE - A longword containing the address of the verb node  
146 0277 1  
147 0278 1 MESSAGE_VECT - The address of a longword to contain the address  
148 0279 1 of a standard message argument vector.  
149 0280 1  
150 0281 1 Implicit Inputs  
151 0282 1  
152 0283 1 None  
153 0284 1  
154 0285 1 Implicit Outputs  
155 0286 1  
156 0287 1 On success, the command execution tree is constructed.  
157 0288 1 On failure, a message argument vector is constructed or obtained.  
158 0289 1  
159 0290 1 Routine value  
160 0291 1  
161 0292 1 STSSK_SUCCESS (1) - Success. Command execution tree constructed.  
162 0293 1 STSSK_SEVERE (4) - Failure. Error encountered. Message argument  
163 0294 1 constructed and returned.  
164 0295 1  
165 0296 1 Side Effects  
166 0297 1  
167 0298 1 None  
168 0299 1!  
169 0300 1  
170 0301 2 BEGIN  
171 0302 2  
172 0303 2 MAP  
173 0304 2 input_desc : REF dbg$stg_desc,  
174 0305 2 verb_node : REF dbg$verb_node;  
175 0306 2  
176 0307 2 BIND  
177 0308 2 dbg$cs_colon = UPLIT BYTE (1, dbg$k_colon),  
178 0309 2 dbg$cs_slash = UPLIT BYTE (1, dbg$k_slash),  
179 0310 2 dbg$cs_comma = UPLIT BYTE (1, dbg$k_comma),  
180 0311 2 dbg$cs_cr = UPLIT BYTE (1, dbg$k_car return),  
181 0312 2 dbg$cs_backslash = UPLIT BYTE (1, dbg$k_backslash);  
182 0313 2  
183 0314 2 LOCAL  
184 0315 2 modrstptr,  
185 0316 2 noun_node : REF dbg$noun_node,  
186 0317 2 prev_noun_node : REF dbg$noun_node,
```

```

187 0318 2 low_lnum,
188 0319 2 high_lnum,
189 0320 2 first_flag;
190 0321 2
191 0322 2
192 0323 2
193 0324 2 ! Create and link a noun node
194 0325 2 noun_node = dbg$get_tempmem(dbg$k_noun_node_size);
195 0326 2 verb_node[dbg$l_verb_object_ptr] = .noun_node;
196 0327 2
197 0328 2 first_flag = TRUE;
198 0329 2 modrstptr = 0;
199 0330 2
200 0331 2 ! Check for TYPE <cr>
201 0332 2
202 0333 2 IF dbg$ncmp(.input_desc, dbg$cs_cr, 1)
203 0334 2 THEN
204 0335 2
205 0336 3 BEGIN ! no argument supplied
206 0337 3 ! Try to get default module.
207 0338 3
208 0339 3
209 0340 3 IF .dbg$src_next_modrstptr EQL 0
210 0341 3 THEN
211 0342 4 BEGIN
212 0343 4 ! report an error
213 0344 4 .message_vect = dbg$make_arg_vect(dbg$nonxtlin);
214 0345 4 RETURN $f$sk_severe;
215 0346 3 END;
216 0347 3
217 0348 3 ! The module rst pointer is placed in the adjective field of
218 0349 3 ! the noun node.
219 0350 3
220 0351 3 noun_node [dbg$l_adjective_ptr] = .dbg$src_next_modrstptr;
221 0352 3
222 0353 3 ! Fill in the line numbers based on global info
223 0354 3
224 0355 3 noun_node [dbg$l_noun_value] = .dbg$src_next_lnum;
225 0356 3 noun_node [dbg$l_noun_value2] = .dbg$src$next_lnum;
226 0357 3
227 0358 3 ! The link field is zero
228 0359 3
229 0360 3 noun_node [dbg$l_noun_link] = 0;
230 0361 3
231 0362 3
232 0363 3
233 0364 3
234 0365 2
235 0366 2
236 0367 2
237 0368 2
238 0369 2
239 0370 2
240 0371 2
241 0372 3 ! name_buf : VECTOR [81, BYTE]; ! Holds counted string with
242 0373 3 ! the module name
243 0374 3

```

```
244 0375 3
245 0376 3
246 0377 3
247 0378 3
248 0379 3
249 0380 3
250 0381 3
251 0382 3
252 0383 3
253 0384 3
254 0385 3
255 0386 3
256 0387 3
257 0388 3
258 0389 3
259 0390 3
260 0391 3
261 0392 3
262 0393 3
263 0394 3
264 0395 3
265 0396 3
266 0397 3
267 0398 3
268 0399 4
269 0400 4
270 0401 4
271 0402 4
272 0403 4
273 0404 4
274 0405 3
275 0406 3
276 0407 3
277 0408 3
278 0409 3
279 0410 3
280 0411 4
281 0412 4
282 0413 4
283 0414 3
284 0415 3
285 0416 3
286 0417 3
287 0418 3
288 0419 3
289 0420 3
290 0421 4
291 0422 4
292 0423 4
293 0424 4
294 0425 4
295 0426 4
296 0427 4
297 0428 4
298 0429 4
299 0430 3
300 0431 3

    LOCAL
        string_ptr,
        length,
        char;

    ! Loop through the list of line number ranges.
    ! The most general form of the command is:
    ! TYPE mod1\l1:h1, mod2\l2:h2, ... , modn\ln:hn
    ! The user may omit the module, in which case it
    ! defaults to a previously-specified module or
    ! to a module determined by the current scope.
    ! He may omit the high line num, in which case it
    ! is the same as the low line num.

    name_buf[0] = 0;
    string_ptr = .input_desc[dsc$A_pointer];
    length = .input_desc[dsc$W_length];
    ! read past leading blanks
    WHILE .length GTR 0
    DO
        BEGIN
            char = ch$Rchar_a(string_ptr);
            length = .length - 1;
            IF .char NEQ dbg$K_blank
            THEN
                EXITLOOP;
            END;

        ! If the length reaches zero then it is an error
        IF .length EQL 0 AND .char EQL dbg$K_blank
        THEN
            BEGIN
                .message_vect = dbg$Nmake_arg_vect(dbg$_needmore);
                RETURN st$S$K_severe;
            END;

        ! Read until we reach a separating character.
        ! Place the characters into name_buf as we read them.

        WHILE .length GTR 0
        DO
            BEGIN
                IF .char EQL '\' OR .char EQL ':'
                OR .char EQL ';' OR .char EQL ','
                THEN
                    EXITLOOP;
                name_buf[0] = .name_buf[0] + 1;
                name_buf[.name_buf[0]] = .char;
                char = ch$Rchar_a(string_ptr);
                length = .length - 1;
            END;
```

```
0432 3 IF .char EQ ' \' ! this signifies that we read a module name
0433 3 THEN
0434 4 BEGIN
0435 4
0436 4 ! Update the input descriptor
0437 4 input_desc[dsc$a_pointer] = .string_ptr;
0438 4 input_desc[dsc$w_length] = .length;
0439 4
0440 4 ! convert the name to an r$ pointer and put it in the
0441 4 ! adjective field.
0442 4
0443 4 noun_node[dbg$1_adjective_ptr] =
0444 4     dbg$sta_getsourcmod(name_buf);
0445 4
0446 4 ! If the above routine returned zero then the user entered
0447 4 ! an invalid module.
0448 4
0449 4 IF .noun_node[dbg$1_adjective_ptr] EQ 0
0450 4 THEN
0451 5 BEGIN
0452 5     .message_vect = dbg$nmake_arg_vect(
0453 5         dbg$nosuchmodu, 1, name_buf);
0454 5     RETURN st$severe;
0455 4 END;
0456 4
0457 4 ! fill in new value of modrstptr
0458 4 modrstptr = .noun_node[dbg$1_adjective_ptr];
0459 4
0460 4 END ! pick up module name
0461 4
0462 3 ELSE
0463 3
0464 4 BEGIN ! fill in default module
0465 4
0466 4 IF .modrstptr NEQ 0
0467 4 THEN
0468 4     ! use global default
0469 4     noun_node[dbg$1_adjective_ptr] = .modrstptr
0470 4
0471 4 ELSE
0472 4
0473 5 BEGIN
0474 5     ! No default in modrstptr, so try to fill
0475 5     ! in a module based on current scope
0476 5
0477 5 modrstptr = dbg$sta_getsourcmod(0);
0478 5
0479 5 IF .modrstptr EQ 0
0480 5 THEN
0481 5
0482 6 BEGIN
0483 6
0484 6 ! If this is still zero, then we have no scope with
0485 6 ! which to supply a module. Report an error.
0486 6
0487 6 .message_vect = dbg$nmake_arg_vect(
0488 6     dbg$noscope, 1, .dbg$src_next_lnum);
```

```
358      0489 6      RETURN sts$k_severe;
359      0490 6
360      0491 6
361      0492 6
362      0493 5
363      0494 5      ! we have found a module.
364      0495 5      ! fill in the adjective field.
365      0496 5      noun_node[dbg$1_adjective_ptr] = .modrptr;
366      0497 5
367      0498 5
368      0499 3      END; ! Fill in default module
369      0500 3
370      0501 3      ! read the low line num
371      0502 3
372      0503 3      IF NOT dbg$nsave_decimal_integer(.input_desc, low_lnum,
373      0504 3      .message_vect)
374      0505 3      THEN
375      0506 3      RETURN sts$k_severe;
376      0507 3
377      0508 3      ! Now look for colon which signifies that the user has also
378      0509 3      ! specified a high line num.
379      0510 3
380      0511 3      IF dbg$nmatch(.input_desc,dbg$cs_colon,1)
381      0512 3      THEN
382      0513 3
383      0514 4      BEGIN
384      0515 4      ! Get high line num
385      0516 4
386      0517 4      IF NOT dbg$nsave_decimal_integer(
387      0518 4      .input_desc, high_lnum, .message_vect)
388      0519 4      THEN
389      0520 4      RETURN sts$k_severe;
390      0521 4
391      0522 4
392      0523 4
393      0524 4
394      0525 3
395      0526 3
396      0527 3      ! high line num same as low line num
397      0528 3      high_lnum = .low_lnum;
398      0529 3
399      0530 3
400      0531 3      ! Fill in the fields of noun_node
401      0532 3
402      0533 3      noun_node[dbg$1_noun_value] = .low_lnum;
403      0534 3      noun_node[dbg$1_noun_value2] = .high_lnum;
404      0535 3
405      0536 3      ! Link in the noun node
406      0537 3
407      0538 3      IF NOT .first_flag
408      0539 3      ! The first noun has already been linked to the verb
409      0540 3      THEN
410      0541 3
411      0542 4      BEGIN
412      0543 4
413      0544 4      prev_noun_node[dbg$1_noun_link] = .noun_node;
414      0545 4      noun_node[dbg$1_noun_link] = 0;
```

```
415      0546 6
416      0547 5
417      0548 5
418      0549 5
419      0550 5
420      0551 5
421      0552 5
422      0553 5
423      0554 5
424      0555 5
425      0556 5
426      0557 5
427      0558 5
428      0559 5
429      0560 5
430      0561 5
431      0562 5
432      0563 5
433      0564 5
434      0565 4
435      0566 4
436      0567 4
437      0568 4
438      0569 4
439      0570 4
440      0571 3
441      0572 3
442      0573 3
443      0574 3
444      0575 2
445      0576 2
446      0577 2
447      0578 2
448      0579 1 END; ! dbg$npars_type
: INFO#250 L1:0544
: Referenced LOCAL symbol PREV_NOUN_NODE is probably not initialized
```

```
.TITLE (DBGNTYPE
.IDENT  \V04-000\
.PSECT  DBG$PLIT,NOWRT, SHR, PIC,0
3A 01 00000 P.AAA: .BYTE 1, 58
2F 01 00002 P.AAB: .BYTE 1, 47
2C 01 00004 P.AAC: .BYTE 1, 44
0D 01 00006 P.AAD: .BYTE 1, 13
5C 01 00008 P.AAE: .BYTE 1, 92
.PSECT  DBG$OWN,NOEXE, PIC,2
00000 NAME_BUF:
.BLK8  81
DBG$CS_COLON=      P.AAA
DBG$CS_SLASH=      P.AAB
DBG$CS_COMMAS=     P.AAC
```

|           |                          |                                                |                                                            |
|-----------|--------------------------|------------------------------------------------|------------------------------------------------------------|
|           |                          |                                                | DBG\$CS_CR= P.AAD                                          |
|           |                          |                                                | DBG\$CS_BACKSLASH= P.AAE                                   |
|           |                          |                                                | .EXTRN DBG\$STA_GETSOURCEMOD                               |
|           |                          |                                                | .EXTRN DBG\$STA_SYMNAME                                    |
|           |                          |                                                | .EXTRN DBG\$PRINT, DBG\$FLUSHBUF                           |
|           |                          |                                                | .EXTRN DBG\$NNEXT WORD, DBG\$NEWLINE                       |
|           |                          |                                                | .EXTRN DBG\$NMATCH, DBG\$GET_TEMP MEM                      |
|           |                          |                                                | .EXTRN DBG\$NMAKE ARG VECT                                 |
|           |                          |                                                | .EXTRN DBG\$NSAVE DECIMAL_INTEGER                          |
|           |                          |                                                | .EXTRN DBG\$NSYNTAX_ERROR                                  |
|           |                          |                                                | .EXTRN DBG\$SCR SCREEN NORMAL                              |
|           |                          |                                                | .EXTRN DBG\$SRC_LINE TO REC                                |
|           |                          |                                                | .EXTRN DBG\$SRC_TYPE LNUM SOURCE                           |
|           |                          |                                                | .EXTRN LIB\$DO COMMAND, LIB\$SPAWN                         |
|           |                          |                                                | .EXTRN SMG\$SET_KEYPAD MODE                                |
|           |                          |                                                | .EXTRN SY\$FAO, DBG\$GB_KEYPAD_INPUT                       |
|           |                          |                                                | .EXTRN DBG\$GL_KEYBOARD_ID                                 |
|           |                          |                                                | .EXTRN DBG\$GV_CONTROL, DBG\$GL_DEVELOPER                  |
|           |                          |                                                | .EXTRN DBG\$GL_SCREEN_SOURCE                               |
|           |                          |                                                | .EXTRN DBG\$SRC_NEXT_MODRSTPTR                             |
|           |                          |                                                | .EXTRN DBG\$SRC_NEXT_LNUM                                  |
|           |                          |                                                | .PSECT DBG\$CODE, NOWRT, SHR, PIC, 0                       |
|           |                          | OFFC 00000                                     | .ENTRY DBG\$NPARSE_TYPE, Save R2,R3,R4,R5,R6,R7,R8,-: 0261 |
|           |                          | 5B 00000000: EF 9E 00002                       | MOVAB DBG\$CS_CR, R11                                      |
|           |                          | 5A 00000000: EF 9E 00009                       | MOVAB NAME BUF, R10                                        |
|           |                          | 5E 08 C2 00010                                 | SUBL2 #8, SP                                               |
|           |                          | 00 04 DD 00013                                 | PUSHL #4                                                   |
| 00000000G | 00 01 FB 00015           | CALLS #1, DBG\$GET_TEMP MEM                    |                                                            |
|           | 52 50 DD 0001C           | MOVL R0, NOUN NODE                             |                                                            |
|           | 50 08 AC DD 0001F        | MOVL VERB_NODE, R0                             |                                                            |
| 08        | A0 52 DD 00023           | MOVL NOUN_NODE, 8(R0)                          |                                                            |
|           | 59 01 DD 00027           | MOVL #1, FIRST_FLAG                            |                                                            |
|           | 57 D4 0002A              | CLRL MODRSTPTR                                 |                                                            |
|           | 01 DD 0002C              | PUSHL #1                                       |                                                            |
|           | 58 DD 0002E              | PUSHL R11                                      |                                                            |
|           | 54 04 AC DD 00030        | MOVL INPUT_DESC, R4                            |                                                            |
|           | 54 DD 00034              | PUSHL R4                                       |                                                            |
| 00000000G | 00 03 FB 00036           | CALLS #3, DBG\$NMATCH                          |                                                            |
|           | 2C 50 E9 0003D           | BLBC R0, 2\$                                   |                                                            |
|           | 00000000G 00 D5 00040    | TSTL DBG\$SRC_NEXT_MODRSTPTR                   |                                                            |
|           | 08 12 00046              | BNEQ 1\$                                       |                                                            |
|           | 00028CEO 8F DD 00048     | PUSHL #167136                                  |                                                            |
|           | 42 11 0004E              | BRB 5\$                                        |                                                            |
| 04        | A2 00000000G 00 DD 00050 | 1\$: MOVL DBG\$SRC_N_T_MODRSTPTR, 4(NOUN_NODE) |                                                            |
|           | 50 00000000G 00 DD 00058 | MOVL DBG\$SRC_NEXT_LNUM, R0                    |                                                            |
|           | 62 50 DD 0005F           | MOVL R0, (NOUN_NODE)                           |                                                            |
| 0C        | A2 50 DD 00062           | MOVL R0, 12(NOUN_NODE)                         |                                                            |
|           | 08 A2 D4 00066           | CLRL 8(NOUN_NODE)                              |                                                            |
|           | 0157 31 00069            | BRW 19\$                                       |                                                            |
|           | 56 04 A4 DD 0006E        | 2\$: CLRB NAME BUF                             |                                                            |
|           | 55 64 3C 00072           | MOVL 4(R4), STRING_PTR                         |                                                            |
|           | 55 D5 00075              | MOVZWL (R4), LENGTH                            |                                                            |
|           | 0A 15 00077              | TSTL LENGTH                                    |                                                            |
|           |                          | BLEQ 4\$                                       |                                                            |

|                       |          |          |                                      |               |        |
|-----------------------|----------|----------|--------------------------------------|---------------|--------|
| 53                    | 86       | 9A 00079 | MOVZBL (STRING_PTR)+, CHAR           | : 0400        |        |
| 20                    | 55       | D7 0007C | DECL LENGTH                          | : 0401        |        |
|                       | 53       | D1 0007E | CMPL CHAR, #32                       | : 0402        |        |
|                       | F2       | 13 00081 | BEQL 3\$                             | : 0409        |        |
|                       | 55       | D5 00083 | 4\$: TSTL LENGTH                     | : 0412        |        |
| 20                    | 15       | 12 00085 | BNEQ 6\$                             | : 0419        |        |
|                       | 53       | D1 00087 | CMPL CHAR, #32                       | : 0422        |        |
|                       | 10       | 12 0008A | BNEQ 6\$                             | : 0423        |        |
| 00000000G 00 000280D0 | 8F       | DD 0008C | PUSHL #164048                        | : 0426        |        |
|                       | 01       | FB 00092 | 5\$: CALLS #1, DBG\$NMAKE_ARG_VECT   | : 0427        |        |
|                       | 0110     | 31 00099 | BRW 16\$                             | : 0428        |        |
|                       | 55       | D5 0009C | 6\$: TSTL LENGTH                     | : 0429        |        |
| 0000005C 8F           | 28       | 15 0009E | BLEQ 7\$                             | : 0432        |        |
|                       | 53       | D1 000A0 | CMPL CHAR, #92                       | : 0437        |        |
|                       | 1F       | 13 000A7 | BEQL 7\$                             | : 0438        |        |
| 2C                    | 53       | D1 000A9 | CMPL CHAR, #44                       | : 0444        |        |
| 3A                    | 1A       | 13 000AC | BEQL 7\$                             | : 0449        |        |
|                       | 53       | D1 000AE | CMPL CHAR, #58                       | : 0452        |        |
|                       | 15       | 13 000B1 | BEQL 7\$                             | : 0458        |        |
| 20                    | 53       | D1 000B3 | CMPL CHAR, #32                       | : 0462        |        |
|                       | 10       | 13 000B6 | BEQL 7\$                             | : 0463        |        |
|                       | 6A       | 96 000B8 | INC8 NAME_BUF                        | : 0466        |        |
| 50                    | 6A       | 9A 000BA | MOVZBL NAME_BUF, R0                  | : 0467        |        |
| 6A40                  | 53       | 90 000BD | MOVB CHAR, NAME_BUF[R0]              | : 0468        |        |
| 53                    | 86       | 9A 000C1 | MOVZBL (STRING_PTR)+, CHAR           | : 0469        |        |
|                       | 55       | D7 000C4 | DECL LENGTH                          | : 0470        |        |
| 0000005C 8F           | D4       | 11 000C6 | BRB 6\$                              | : 0473        |        |
|                       | 53       | D1 000C8 | 7\$: CMPL CHAR, #92                  | : 0476        |        |
|                       | 28       | 12 000CF | BNEQ 9\$                             | : 0477        |        |
| 04 A4                 | 56       | D0 000D1 | MOVL STRING_PTR, 4(R4)               | : 0478        |        |
| 64                    | 55       | B0 000D5 | MOVW LENGTH, (R4)                    | : 0479        |        |
| 00000000G 00 04 A2    | 5A       | DD 000D8 | PUSHL R10                            | : 0480        |        |
|                       | 01       | FB 000DA | CALLS #1, DBG\$STA_GETSOURCEMOD      | : 0481        |        |
|                       | 50       | D0 000E1 | MOVL R0, 4(NOUN_NODE)                | : 0482        |        |
|                       | 0C       | 12 000E5 | BNEQ 8\$                             | : 0483        |        |
|                       | 5A       | DD 000E7 | PUSHL R10                            | : 0484        |        |
|                       | 01       | DD 000E9 | PUSHL #1                             | : 0485        |        |
|                       | 000281E8 | 8F       | D0 000EB                             | PUSHL #164328 | : 0486 |
|                       | 26       | 11 000F1 | BRB 10\$                             | : 0487        |        |
| 57 04                 | A2       | D0 000F3 | 8\$: MOVL 4(NOUN_NODE), MODRSTPTR    | : 0488        |        |
|                       | 2E       | 11 000F7 | BRB 12\$                             | : 0489        |        |
|                       | 57       | D5 000F9 | 9\$: TSTL MODRSTPTR                  | : 0490        |        |
|                       | 26       | 12 000FB | BNEQ 11\$                            | : 0491        |        |
| 00000000G 00 57       | 7E       | D4 000FD | CLRL -(SP)                           | : 0492        |        |
|                       | 01       | FB 000FF | CALLS #1, DBG\$STA_GETSOURCEMOD      | : 0493        |        |
|                       | 50       | D0 00106 | MOVL R0, MODRSTPTR                   | : 0494        |        |
|                       | 18       | 12 00109 | BNEQ 11\$                            | : 0495        |        |
| 00000000G 00          | 00       | DD 0010B | PUSHL DBG\$SRC_NEXT_LNUM             | : 0496        |        |
|                       | 01       | DD 00111 | PUSHL #1                             | : 0497        |        |
| 00000000G 00 00028972 | 8F       | DD 00113 | PUSHL #166258                        | : 0498        |        |
|                       | 03       | FB 00119 | 10\$: CALLS #3, DBG\$NMAKE_ARG_VECT  | : 0499        |        |
| 04 A2                 | 0089     | 31 00120 | BRW 16\$                             | : 0500        |        |
|                       | 57       | D0 00123 | 11\$: MOVL MODRSTPTR, 4(NOUN_NODE)   | : 0501        |        |
|                       | 0C       | AC 00127 | 12\$: PUSHL MESSAGE_VECT             | : 0502        |        |
|                       | 04       | AE 0012A | PUSHAB LOW_LNUM                      | : 0503        |        |
| 00000000G 00          | 54       | DD 0012D | PUSHL R4                             | : 0504        |        |
|                       | 03       | FB 0012F | CALLS #3, DBG\$NSAVE_DECIMAL_INTEGER | : 0505        |        |

|           |    |      |          |          |                                |                           |      |
|-----------|----|------|----------|----------|--------------------------------|---------------------------|------|
| 00000000G | 00 | 77   | 50       | E9 00136 | BLBC                           | R0, 17\$                  | 0511 |
|           | 14 | FA   | 01       | DD 00139 | PUSHL                          | #1                        |      |
|           |    |      | AB       | 9F 0013B | PUSHAB                         | DBG\$CS_COLON             |      |
|           |    |      | 54       | DD 0013E | PUSHL                          | R4                        |      |
|           |    |      | 03       | FB 00140 | CALLS                          | #3, DBG\$NMATCH           |      |
|           |    |      | 50       | E9 00147 | BLBC                           | R0, 13\$                  |      |
|           |    | 0C   | AC       | DD 0014A | PUSHL                          | MESSAGE_VECT              | 0519 |
|           |    | 08   | AE       | 9F 0014D | PUSHAB                         | HIGH_LNUM                 | 0518 |
|           |    |      | 54       | DD 00150 | PUSHL                          | R4                        | 0519 |
| 00000000G | 00 | 03   | FB 00152 | CALLS    | #3, DBG\$NSAVE_DECIMAL_INTEGER |                           |      |
|           | 06 | 50   | E8 00159 | BLBS     | R0, 14\$                       |                           |      |
|           |    |      | 52       | 11 0015C | BRB                            | 17\$                      | 0521 |
| 04        | AE | 6E   | DO 0015E | 13\$:    | MOVL                           | LOW_LNUM, HIGH_LNUM       | 0529 |
|           | 62 | 6E   | DO 00162 | 14\$:    | MOVL                           | LOW_LNUM, (NOUN_NODE)     | 0533 |
| 0C        | A2 | 04   | AE       | DO 00165 | MOVL                           | HIGH_LNUM, 12(NOUN_NODE)  | 0534 |
|           | 07 | 59   | E8 0016A | BLBS     | FIRST_FLAG, 15\$               | 0538                      |      |
| 08        | A8 | 52   | DO 0016D | MOVL     | NOUN_NODE, 8(PREV_NOUN_NODE)   | 0544                      |      |
|           |    | 08   | A2       | D4 00171 | CLRL                           | 8(NOUN_NODE)              | 0545 |
|           |    | 58   | 59       | D4 00174 | 15\$:                          | FIRST_FLAG                | 0550 |
|           |    |      | 52       | DO 00176 | MOVL                           | NOUN_NODE, PREV_NOUN_NODE | 0551 |
|           |    |      | 01       | DD 00179 | PUSHL                          | #1                        | 0555 |
| 00000000G | 00 | 0810 | 8F       | BB 0017B | PUSHR                          | #^M<R4, R11>              |      |
|           | 3A | 03   | FB 0017F | CALLS    | #3, DBG\$NMATCH                |                           |      |
|           |    | 50   | E8 00186 | BLBS     | R0, 19\$                       |                           | 0562 |
|           |    |      | 01       | DD 00189 | PUSHL                          | #1                        |      |
|           |    | FE   | AB       | 9F 0018B | PUSHAB                         | DBG\$CS_COMMAS            |      |
|           |    |      | 54       | DD 0018E | PUSHL                          | R4                        |      |
| 00000000G | 00 | 03   | FB 00190 | CALLS    | #3, DBG\$NMATCH                |                           |      |
|           | 1A | 50   | E8 00197 | BLBS     | R0, 18\$                       |                           | 0568 |
| 00000000G | 00 | 54   | DD 0019A | PUSHL    | R4                             |                           |      |
|           |    | 01   | FB 0019C | CALLS    | #1, DBG\$NNEXT_WORD            |                           |      |
| 00000000G | 00 | 50   | DD 001A3 | PUSHL    | R0                             |                           |      |
| 0C        | J0 | 01   | FB 001A5 | CALLS    | #1, DBG\$NSYNTAX_ERROR         |                           |      |
|           | BC | 50   | DO 001AC | 16\$:    | MOVL                           | R0, @MESSAGE_VECT         |      |
|           | 50 | 04   | DO 001B0 | 17\$:    | MOVL                           | #4, R0                    | 0569 |
|           |    |      | 04       | 001B3    | RET                            |                           |      |
| 00000000G | 00 | 04   | DD 001B4 | 18\$:    | PUSHL                          | #4                        | 0574 |
|           | 52 | 01   | FB 001B6 | CALLS    | #1, DBG\$GET_TEMP MEM          |                           |      |
|           |    | 50   | DO 001BD | MOVL     | R0, NOUN_NODE                  |                           | 0367 |
|           |    | FEA9 | 31       | 001C0    | BRW                            | 2\$                       | 0577 |
|           | 50 | 01   | DO 001C3 | 19\$:    | MOVL                           | #1, R0                    | 0579 |
|           |    |      | 04       | 001C6    | RET                            |                           |      |

; Routine Size: 455 bytes, Routine Base: DBG\$CODE + 0000

449 0580 1  
450 0581 1

```
452 0582 1 GLOBAL ROUTINE DBG$NEXECUTE_TYPE(VERB_NODE, MESSAGE_VECT) =  
453 0583 1  
454 0584 1 Functional Description  
455 0585 1  
456 0586 1 This routine performs the action associated with the TYPE  
457 0587 1 command.  
458 0588 1  
459 0589 1 Formal Parameters  
460 0590 1  
461 0591 1 VERB_NODE - A longword containing the address of the  
462 0592 1 head (verb) node.  
463 0593 1  
464 0594 1 MESSAGE_VECT - The address of a longword to contain the  
465 0595 1 address of an error message vector  
466 0596 1  
467 0597 1 Implicit Inputs  
468 0598 1  
469 0599 1 The command tree contains a verb node and a linked list  
470 0600 1 of one or more noun nodes.  
471 0601 1  
472 0602 1 Implicit Outputs  
473 0603 1  
474 0604 1 This routine calls a routine in DBGSOURCE which displays the  
475 0605 1 source lines to the user.  
476 0606 1  
477 0607 1 Routine Value  
478 0608 1  
479 0609 1 A completion code.  
480 0610 1  
481 0611 1 Completion Codes  
482 0612 1  
483 0613 1 STSS$K_SUCCESS (1) - Success. Command executed  
484 0614 1 STSS$K_SEVERE (4) - Failure. The command could not be  
485 0615 1 executed. An error message is constructed.  
486 0616 1  
487 0617 1 Side Effects  
488 0618 1  
489 0619 1 None  
490 0620 1  
491 0621 1  
492 0622 2 BEGIN  
493 0623 2  
494 0624 2 MAP  
495 0625 2 verb_node : REF dbg$verb_node;  
496 0626 2  
497 0627 2 LOCAL  
498 0628 2 noun_node : REF dbg$noun_node,  
499 0629 2 modnameptr;  
500 0630 2  
501 0631 2 noun_node = .verb_node[dbg$1_verb_object_ptr];  
502 0632 2  
503 0633 2 DO  
504 0634 3 BEGIN  
505 0635 3  
506 0636 3 ! Loop through linked list of noun nodes.  
507 0637 3  
508 0638 3 ! First output the module name.
```

```

509 0639 3
510 0640 3
511 0641 3
512 0642 3
513 0643 3
514 0644 3
515 0645 3
516 0646 3
517 0647 3
518 0648 3
519 0649 3
520 0650 3
521 0651 4
522 0652 4
523 0653 4
524 0654 4
525 0655 3
526 0656 3
527 0657 3
528 0658 3
529 0659 3
530 0660 3
531 0661 3
532 0662 3
533 0663 3
534 0664 3
535 0665 3
536 0666 2
537 0667 2
538 0668 2
539 0669 2
540 0670 1

      ! Set up for FAO_PUT.
      DBG$FLUSHBUF();

      ! Get the module name and print it out. Note that the printing of
      ! the module name is suppressed if the source line output is being
      ! directed to a source screen display.

      IF .DBG$GL_SOURCE_EQL 0
      THEN
          BEGIN
              DBG$STA_SYMNAME(.NOUN_NODE[DBG$L_ADJECTIVE_PTR], MODNAMEPTR);
              DBG$PRINT(UPLIT BYTE(%ASCIC 'module !AC'), .MODNAMEPTR);
              DBG$NEWLINE();
          END;

      ! Call the routine which actually displays the sour    lines.

      DBG$SRC_TYPE_LNUM_SOURCE(.NOUN_NODE[DBG$L_ADJECTIVE_PTR],
                               .NOUN_NODE[DBG$L_NOUN_VALUE], 0,
                               .NOUN_NODE[DBG$L_NOUN_VALUE2], 0,
                               0, FALSE);

      END

      UNTIL 0 EQL (NOUN_NODE = .NOUN_NODE[DBG$L_NOUN_LINK]);

      RETURN STSSK_SUCCESS

      END;

```

43 41 21 20 65 6C 75 64 6F 6D 0A 0000A P.AAF: .ASCII <10>\module !AC\ .PSECT DBGSPLIT,NOWRT, SHR, PIC,0

|  |  |  |  | .PSECT | DBG\$CODE, NOWRT, SHR, PIC, 0 |        |
|--|--|--|--|--------|-------------------------------|--------|
|  |  |  |  | .ENTRY | DBG\$NEXECUTE_TYPE, Save R2   | : 0582 |
|  |  |  |  | SUBL2  | #4, SP                        |        |
|  |  |  |  | MOVL   | VERB NODE, R0                 | : 0631 |
|  |  |  |  | MOVL   | 8(ROT), NOUN NODE             |        |
|  |  |  |  | CALLS  | #0, DBG\$FLUSHBUF             | : 0642 |
|  |  |  |  | TSTL   | DBG\$GL_SCREEN_SOURCE         | : 0649 |
|  |  |  |  | BNEQ   | 2\$                           |        |
|  |  |  |  | PUSHL  | SP                            | : 0652 |
|  |  |  |  | PUSHL  | 4(NOUN NODE)                  |        |
|  |  |  |  | CALLS  | #2, DBG\$STA_SYMNAME          |        |
|  |  |  |  | PUSHL  | MODNAMEPTR                    | : 0653 |
|  |  |  |  | PUSHAB | P.AAF                         |        |
|  |  |  |  | CALLS  | #2, DBG\$PRINT                |        |
|  |  |  |  | CALLS  | #0, DBG\$NEWLINE              | : 0654 |

|           |    |    |               |       |                               |      |
|-----------|----|----|---------------|-------|-------------------------------|------|
|           |    | 7E | 7C 0003E 2\$: | CLRQ  | -(SP)                         | 0660 |
|           |    | 7E | D4 00040      | CLRL  | -(SP)                         | 0662 |
|           | 0C | A2 | DD 00042      | PUSHL | 12(NOUN_NODE)                 | 0660 |
|           |    | 7E | D4 00045      | CLRL  | -(SP)                         | 0660 |
|           |    | 62 | DD 00047      | PUSHL | (NOUN_NODE)                   | 0661 |
| 00000000G | 00 | 04 | A2 DD 00049   | PUSHL | 4(NOUN_NODE)                  | 0660 |
|           | 52 | 08 | 07 FB 0004C   | CALLS | #7, DBG\$SRC_TYPE_LNUM_SOURCE | 0666 |
|           |    | A2 | DD 00053      | MOVL  | 8(NOUN_NODE), NOUN_NODE       | 0668 |
|           |    | B4 | 12 00057      | BNEQ  | 1\$                           | 0670 |
|           | 50 |    | 01 DD 00059   | MOVL  | #1, R0                        |      |
|           |    |    | 04 0005C      | RET   |                               |      |

; Routine Size: 93 bytes, Routine Base: DBG\$CODE + 01C7

542 0671 1 GLOBAL ROUTINE DBGSNPARSE\_EDIT (INPUT\_DESC, VERB\_NODE, MESSAGE\_VECT) =  
543 0672 1  
544 0673 1 Functional Description  
545 0674 1  
546 0675 1 Parse routine for the EDIT command. EDIT is the DEBUG interface  
547 0676 1 to the EDITH editor.  
548 0677 1  
549 0678 1 The routine DBGSNPARSE\_CMD in DBGSNPARSE has already recognized  
550 0679 1 the "EDIT" verb and has called this routine to parse the  
551 0680 1 rest of the command.  
552 0681 1  
553 0682 1 This routine takes a verb node for the EDIT verb, and a string  
554 0683 1 descriptor for the remaining (unparsed) input.  
555 0684 1  
556 0685 1 The rest of the command execution tree is constructed.  
557 0686 1  
558 0687 1 Formal Parameters  
559 0688 1  
560 0689 1 INPUT\_DESC - A longword containing the address of the  
561 0690 1 command input descriptor.  
562 0691 1  
563 0692 1 VERB\_NODE - A longword containing the address of the verb node  
564 0693 1  
565 0694 1 MESSAGE\_VECT - The address of a longword to contain the address  
566 0695 1 of a standard message argument vector.  
567 0696 1  
568 0697 1 Implicit Inputs  
569 0698 1  
570 0699 1  
571 0700 1  
572 0701 1  
573 0702 1  
574 0703 1  
575 0704 1  
576 0705 1  
577 0706 1  
578 0707 1  
579 0708 1  
580 0709 1  
581 0710 1  
582 0711 1  
583 0712 1  
584 0713 1  
585 0714 1  
586 0715 1  
587 0716 1  
588 0717 2  
589 0718 2  
590 0719 2  
591 0720 2  
592 0721 2  
593 0722 2  
594 0723 2  
595 0724 2  
596 0725 2  
597 0726 2  
598 0727 2  
0671 1  
0672 1  
0673 1  
0674 1  
0675 1  
0676 1  
0677 1  
0678 1  
0679 1  
0680 1  
0681 1  
0682 1  
0683 1  
0684 1  
0685 1  
0686 1  
0687 1  
0688 1  
0689 1  
0690 1  
0691 1  
0692 1  
0693 1  
0694 1  
0695 1  
0696 1  
0697 1  
0698 1  
0699 1  
0700 1  
0701 1  
0702 1  
0703 1  
0704 1  
0705 1  
0706 1  
0707 1  
0708 1  
0709 1  
0710 1  
0711 1  
0712 1  
0713 1  
0714 1  
0715 1  
0716 1  
0717 2  
0718 2  
0719 2  
0720 2  
0721 2  
0722 2  
0723 2  
0724 2  
0725 2  
0726 2  
0727 2  
None  
None  
On success, the command execution tree is constructed.  
On failure, a message argument vector is constructed or obtained.  
STSSK\_SUCCESS (1) - Success. Command execution tree constructed.  
STSSK\_SEVERE (4) - Failure. Error encountered. Message argument  
constructed and returned.  
None  
BEGIN  
MAP  
INPUT\_DESC : REF DBGSSTG\_DESC,  
VERB\_NODE : REF DBGSVERB\_NODE;  
BIND  
DBG\$CS\_EXIT = UPLIT BYTE (4, 'EXIT')  
DBG\$CS\_NOEXIT = UPLIT BYTE (6, 'NOEXIT')  
DBG\$CS\_BACKSLASH = UPLIT BYTE (1, DBGSK\_BACKSLASH),  
DBG\$CS\_CR = UPLIT BYTE (1, DBGSK\_CAR\_RETURN),

```

599 0728 2           DBG$CS_SLASH           = UPLIT BYTE (1, DBG$K_SLASH);
600 0729 2
601 0730 2
602 0731 2
603 0732 2
604 0733 2
605 0734 2
606 0735 2
607 0736 2
608 0737 2
609 0738 2
610 0739 2
611 0740 2
612 0741 2
613 0742 2
614 0743 2
615 0744 2
616 0745 2
617 0746 2
618 0747 2
619 0748 2
620 0749 2
621 0750 2
622 0751 2
623 0752 2
624 0753 2
625 0754 2
626 0755 2
627 0756 3
628 0757 3
629 0758 3
630 0759 3
631 0760 3
632 0761 3
633 0762 3
634 0763 3
635 0764 3
636 0765 3
637 0766 3
638 0767 3
639 0768 3
640 0769 3
641 0770 3
642 0771 3
643 0772 3
644 0773 3
645 0774 2
646 0775 2
647 0776 2
648 0777 2
649 0778 2
650 0779 2
651 0780 2
652 0781 2
653 0782 2
654 0783 2
655 0784 2

           LOCAL
           CHAR,
           DLINÉ_PTR: REF DBG$DLINE_ENTRY,
           DROW,
           LENGTH,
           LINENO,
           LINUM,
           MODRS PTR,
           NAME_BUF: VECTOR [81, BYTE],
           NOUN_NODE: REF DBG$NOUN_NODE,
           RBEG,
           RLEN,
           SRC_DISP: REF DBG$DISP_ENTRY,
           STRING_PTR;

           ! Create and link a noun node.
           ! NOUN_NODE = DBG$GET_TEMP MEM(DBG$K_NOUN_NODE_SIZE);
           ! VERB_NODE[DBG$L_VERB_OBJECT_PTR] = .NOUN_NODE;

           ! Pick up optional /EXIT or /NOEXIT. /NOEXIT is the default.
           ! VERB_NODE[DBG$B_VERB_COMPOSITE] = EDIT_NOEXIT;
           ! WHILE DBG$NMATCH (.INPUT_DESC, DBG$CS_SLASH, 1) DO
           ! BEGIN
           !     SELECT ONE TRUE OF
           !         SET
           !             [DBG$NMATCH (.INPUT_DESC, DBG$CS_EXIT, 1)]:
           !                 VERB_NODE[DBG$B_VERB_COMPOSITE] = EDIT_EXIT;
           !             [DBG$NMATCH (.INPUT_DESC, DBG$CS_NOEXIT, 1)]:
           !                 VERB_NODE[DBG$B_VERB_COMPOSITE] = EDIT_NOEXIT;
           !             [DBG$NMATCH (.INPUT_DESC, DBG$CS_CR, 1)]:
           !                 SIGNAL(DBGS_NEEDMORE);
           !             [OTHERWISE]:
           !                 SIGNAL(DBGS_CMDSYNERR, 1, DBG$NNEXT_WORD(.INPUT_DESC));
           !             TES;
           ! END;

           ! Initialize module and line number.
           ! MODRSTPTR = 0;
           ! LINENO = 0;

           ! Check for EDIT <cr>

```

```
656 0785 2 IF DBG$NMATCH(.INPUT_DESC, DBG$CS_CR, 1)
657 0786 2 THEN
658 0787 3 BEGIN
659 0788 3
660 0789 3
661 0790 3
662 0791 3
663 0792 3
664 0793 3
665 0794 3
666 0795 3
667 0796 3
668 0797 3
669 0798 3
670 0799 4
671 0800 4
672 0801 4
673 0802 4
674 0803 4
675 0804 4
676 0805 4
677 0806 4
678 0807 4
679 0808 4
680 0809 4
681 0810 4
682 0811 4
683 0812 4
684 0813 4
685 0814 3
686 0815 3
687 0816 4
688 0817 4
689 0818 4
690 0819 3
691 0820 3
692 0821 3
693 0822 3
694 0823 2
695 0824 3
696 0825 3
697 0826 3
698 0827 3
699 0828 3
700 0829 3
701 0830 3
702 0831 3
703 0832 3
704 0833 3
705 0834 3
706 0835 3
707 0836 3
708 0837 3
709 0838 4
710 0839 4
711 0840 4
712 0841 4

IF DBG$NMATCH(.INPUT_DESC, DBG$CS_CR, 1)
THEN
BEGIN
| Try to get default module and line number. If screen mode is set,
| then get this info from the source display on the screen. Otherwise,
| use defaulting rules based on the most recent TYPE command. If we
| are unable to default module name and line number then leave
| zeros in the variables to indicate this.
SRC_DISP = .DBG$GL_SCREEN_SOURCE;
IF .SRC_DISP NEQ 0
THEN
BEGIN
MODRSTPTR = .SRC_DISP[DBG$L_DISP_MODPTR];

| Do the computation to figure out what line number is
| the center of the source display.
RBEG = .SRC_DISP[DBG$W_DISP_RBEG];
RLEN = .SRC_DISP[DBG$W_DISP_RLEN];
DROW = .SRC_DISP[DBG$W_DISP_DROW];
DLINE_PTR = .SRC_DISP[DBG$L_DISP_START_LINE_PTR];
LINUM = .DLINE_PTR[DBG$L_DLINE_LNUM];
LINENO = (.DROW-.RBEG) + .RLEN/2 + .LINUM;
END

ELSE IF .DBG$SRC_NEXT_MODRSTPTR NEQ 0
THEN
BEGIN
MODRSTPTR = .DBG$SRC_NEXT_MODRSTPTR;
LINENO = .DBG$SRC_NEXT_LNUM - 1;
END;

ELSE
BEGIN
| Initialize the buffer which will hold the module name.
| Set up to scan the input.
name_buf[0] = 0;
string_ptr = .input_desc[dsc$sa_pointer];
length = .input_desc[dsc$w_length];

| Read past leading blanks
WHILE .length GT 0 DO
BEGIN
char = ch$rchar_a(string_ptr);
length = .length - 1;
IF .char NEQ dbg$k_blank
```

```
713 0842 4
714 0843 4
715 0844 3
716 0845 3
717 0846 3
718 0847 3
719 0848 3
720 0849 3
721 0850 3
722 0851 3
723 0852 3
724 0853 3
725 0854 3
726 0855 3
727 0856 3
728 0857 3
729 0858 4
730 0859 4
731 0860 4
732 0861 4
733 0862 4
734 0863 4
735 0864 4
736 0865 4
737 0866 4
738 0867 4
739 0868 4
740 0869 3
741 0870 3
742 0871 3
743 0872 3
744 0873 3
745 0874 3
746 0875 3
747 0876 4
748 0877 4
749 0878 4
750 0879 4
751 0880 4
752 0881 4
753 0882 4
754 0883 4
755 0884 4
756 0885 4
757 0886 4
758 0887 4
759 0888 4
760 0889 4
761 0890 4
762 0891 4
763 0892 4
764 0893 4
765 0894 4
766 0895 4
767 0896 4
768 0897 4
769 0898 3

    THEN EXITLOOP;
    END;

    ! If the length reaches zero then it is an error.
    IF .length EQ 0
    THEN $DBG_ERROR('DBGNTYPE\DBG$NPARSE_EDIT');

    ! Read until we reach a separating character.
    ! Place the characters into name_buf as we read them.
    WHILE .length GTR 0  DO
        BEGIN
            IF .char EQ '\' OR .char EQ ' '
            THEN
                EXITLOOP;
            name_buf[0] = name_buf[0] + 1;
            IF .name_buf[0] GTR 80
            THEN
                SIGNAL(dbg$cmdsyntax, 1, dbg$next_word(.input_desc));
            name_buf[name_buf[0]] = .char;
            char = ch$rcchar a(string_ptr);
            length = .length - 1;
        END;

    ! Backslash signifies that we just read a module name.
    IF .char EQ '\''
    THEN
        BEGIN

            ! Update the input descriptor
            input_desc[dsc$pointer] = .string_ptr;
            input_desc[dsc$w_length] = .length;

            ! Convert the name to an rst pointer.
            modrstptr = dbg$sta_getsourcmod(name_buf);

            ! If the above routine returned zero then the user entered
            ! an invalid module.
            IF .modrstptr EQ 0
            THEN
                SIGNAL(dbg$nosuchmodu, 1, name_buf);
            END

        ELSE
```

```
770 0899 4 BEGIN
771 0900 4
772 0901 4
773 0902 4
774 0903 4
775 0904 4
776 0905 4
777 0906 4
778 0907 4
779 0908 4
780 0909 4
781 0910 4
782 0911 4
783 0912 4
784 0913 4
785 0914 4
786 0915 4
787 0916 4
788 0917 4
789 0918 3
790 0919 3
791 0920 3
792 0921 3
793 0922 3
794 0923 3
795 0924 3
796 0925 3
797 0926 2
798 0927 2
799 0928 2
800 0929 2
801 0930 2
802 0931 2
803 0932 2
804 0933 2
805 0934 2
806 0935 2
807 0936 2
808 0937 2
809 0938 2
810 0939 1

    ! No module was specified, so try to fill
    ! in a module based on (1) module in current
    ! source display, or (2) module given by previous
    ! TYPE command, or (3) current scope
    SRC_DISP = .DBG$GL_SOURCE;
    IF .SRC_DISP NEQ 0
    THEN
        MODRSTPTR = .SRC_DISP[DBG$L_DISP_MODPTR]
    ELSE IF .DBG$SRC_NEXT_MODRSTPTR NEQ 0
    THEN
        MODRSTPTR = .DBG$SRC_NEXT_MODRSTPTR
    ELSE
        modrstptr = dbg$sta_getsourcmod(0);
    END;

    ! Read the line number.
    IF NOT dbg$nsave_decimal_integer(.input_desc, Lineno, .message_vect)
    THEN
        RETURN sts$k_severe;
    END;

    ! Fill in the fields of the noun node with the module and
    ! line number.
    NOUN_NODE[DBG$L_ADJECTIVE_PTR] = .MODRSTPTR;
    NOUN_NODE[DBG$L_NOUN_VALUE] = .LINENO;

    ! We have finished parsing the EDIT command. Return success.
    RETURN STS$K_SUCCESS;
END;
```

.PSECT DBG\$PLIT,NOWRT, SHR, PIC,0

|                               |                                            |                                             |
|-------------------------------|--------------------------------------------|---------------------------------------------|
| 54 49 58 45 04                | 00015 P.AAG:                               | .BYTE 4                                     |
| 54 49 58 45 06                | 00016 P.AAH:                               | .ASCII \EXIT\                               |
| 54 49 58 45 4F                | 0001A P.AAH:                               | .BYTE 6                                     |
| 54 49 58 45 4E                | 0001B P.AAI:                               | .ASCII \NOEXIT\                             |
| 5C 01                         | 00021 P.AAJ:                               | .BYTE 1, 92                                 |
| 0D 01                         | 00023 P.AAJ:                               | .BYTE 1, 13                                 |
| 2F 01                         | 00025 P.AAK:                               | .BYTE 1, 47                                 |
| 4E 24 47 42 44                | 5C 45 50 59 54 4E 47 42 44 18 00027 P.AAL: | .ASCII <24>\DBGNTYPE\<92>\DBG\$NPARSE_EDIT\ |
| 54 49 44 45 5F 45 53 52 41 50 | 00036                                      |                                             |

DBG\$CS\_EXIT= P.AAG

DBG\$CS\_NOEXIT= P.AAH  
 DBG\$CS\_BACKSLASH= P.AAI  
 DBG\$CS\_CR= P.AAJ  
 DBG\$CS\_SLASH= P.AAK

|              |  |  |  | .PSECT | DBG\$CODE, NOWRT, SHR, PIC, 0                              |      |
|--------------|--|--|--|--------|------------------------------------------------------------|------|
|              |  |  |  | .ENTRY | DBG\$NPARSE_EDIT, Save R2, R3, R4, R5, R6, R7, R8, -; 0671 |      |
|              |  |  |  | MOVAB  | DBGSNMATCH, R11                                            |      |
| 00000000G 00 |  |  |  | MOVAB  | DBG\$CS_CR, R10                                            |      |
|              |  |  |  | MOVAB  | -88(SPT), SP                                               |      |
|              |  |  |  | PUSHL  | #4                                                         |      |
|              |  |  |  | CALLS  | #1, DBG\$GET_TEMPMEM                                       |      |
| 00000000G 08 |  |  |  | MOVL   | R0, NOUN_NODE                                              |      |
|              |  |  |  | MOVL   | VERB_NODE, R2                                              |      |
| 01 A2        |  |  |  | MOVL   | NOUN_NODE, 8(R2)                                           |      |
|              |  |  |  | MOVB   | #2, T(R2)                                                  |      |
| 01 A2        |  |  |  | MOVL   | INPUT_DESC, R4                                             |      |
|              |  |  |  | PUSHL  | #1                                                         |      |
| 01 A2        |  |  |  | PUSHAB | DBG\$CS_SLASH                                              |      |
|              |  |  |  | PUSHL  | R4                                                         |      |
| 01 A2        |  |  |  | CALLS  | #3, DBGSNMATCH                                             |      |
|              |  |  |  | BLBC   | R0, 5\$                                                    |      |
| 01 A2        |  |  |  | PUSHL  | #1                                                         |      |
|              |  |  |  | PUSHAB | DBG\$CS_EXIT                                               |      |
| 01 A2        |  |  |  | PUSHL  | R4                                                         |      |
|              |  |  |  | CALLS  | #3, DBGSNMATCH                                             |      |
| 01 A2        |  |  |  | CMPL   | R0, #1                                                     |      |
|              |  |  |  | BNEQ   | 2\$                                                        |      |
| 01 A2        |  |  |  | MOVB   | #1, 1(R2)                                                  |      |
|              |  |  |  | BRB    | 1\$                                                        |      |
| 01 A2        |  |  |  | PUSHL  | #1                                                         |      |
|              |  |  |  | PUSHAB | DBG\$CS_NOEXIT                                             |      |
| 01 A2        |  |  |  | PUSHL  | R4                                                         |      |
|              |  |  |  | CALLS  | #3, DBGSNMATCH                                             |      |
| 01 A2        |  |  |  | CMPL   | R0, #1                                                     |      |
|              |  |  |  | BNEQ   | 3\$                                                        |      |
| 01 A2        |  |  |  | MOVB   | #2, 1(R2)                                                  |      |
|              |  |  |  | BRB    | 1\$                                                        |      |
| 01 A2        |  |  |  | PUSHL  | #1                                                         |      |
|              |  |  |  | PUSHR  | #^M<R4, R10>                                               |      |
| 00000000G 00 |  |  |  | CALLS  | #3, DBGSNMATCH                                             |      |
|              |  |  |  | CMPL   | R0, #1                                                     |      |
| 00000000G 00 |  |  |  | BNEQ   | 4\$                                                        |      |
|              |  |  |  | PUSHL  | #164048                                                    |      |
| 00000000G 00 |  |  |  | CALLS  | #1, LIB\$SIGNAL                                            |      |
|              |  |  |  | BRB    | 1\$                                                        |      |
| 00000000G 00 |  |  |  | PUSHL  | R4                                                         |      |
|              |  |  |  | CALLS  | #1, DBGSNNEXT_WORD                                         |      |
| 00000000G 00 |  |  |  | PUSHL  | R0                                                         |      |
|              |  |  |  | PUSHL  | #1                                                         |      |
| 00000000G 00 |  |  |  | PUSHL  | #167608                                                    |      |
|              |  |  |  | CALLS  | #3, LIB\$SIGNAL                                            |      |
| 00000000G 00 |  |  |  | BRB    | 1\$                                                        |      |
|              |  |  |  | CLRL   | MODRSTPTR                                                  |      |
|              |  |  |  |        |                                                            | 0755 |
|              |  |  |  |        |                                                            | 0779 |

|  |  |  |      |           |          |        |                                 |                     |      |  |
|--|--|--|------|-----------|----------|--------|---------------------------------|---------------------|------|--|
|  |  |  | 6E   | D4        | 000A2    | CLRL   | LINENO                          | 0780                |      |  |
|  |  |  | 01   | DD        | 000A4    | PUSHL  | #1                              | 0785                |      |  |
|  |  |  | 0410 | 8F        | BB       | PUSHR  | #^M<R4, R10>                    |                     |      |  |
|  |  |  | 6B   | 03        | FB       | CALLS  | #3, DBG\$NMATCH                 |                     |      |  |
|  |  |  | 47   | 50        | E9       | BLBC   | R0, 8\$                         |                     |      |  |
|  |  |  | 52   | 00000000G | 00       | MOVL   | DBG\$GL_SCREEN_SOURCE, SRC_DISP | 0796                |      |  |
|  |  |  | 27   | 13        | 000B7    | BEQL   | 6\$                             | 0797                |      |  |
|  |  |  | 56   | 34        | A2       | MOVL   | 52(SRC_DISP), MODRSTPTR         | 0800                |      |  |
|  |  |  | 55   | 10        | A2       | MOVZWL | 16(SRC_DISP), RBEG              | 0806                |      |  |
|  |  |  | 50   | 12        | A2       | MOVZWL | 18(SRC_DISP), RLEN              | 0807                |      |  |
|  |  |  | 51   | 18        | A2       | MOVZWL | 24(SRC_DISP), DROW              | 0808                |      |  |
|  |  |  | 53   | 20        | A2       | MOVL   | 32(SRC_DISP), DLINE_PTR         | 0809                |      |  |
|  |  |  | 53   | 10        | A3       | MOVL   | 16(DLINE_PTR), LINUM            | 0810                |      |  |
|  |  |  | 51   | 55        | C2       | SUBL2  | RBEG, R1                        | 0811                |      |  |
|  |  |  | 50   | 02        | C6       | DIVL2  | #2, R0                          |                     |      |  |
|  |  |  | 50   | 51        | C0       | ADDL2  | R1, R0                          |                     |      |  |
|  |  |  | 6E   | 50        | 53       | ADDL3  | LINUM, R0, LINENO               |                     |      |  |
|  |  |  |      | 53        | C1       | BRB    | 7\$                             | 0797                |      |  |
|  |  |  |      | 14        | 11       | MOVL   | DBG\$SRC_NEXT_MODRSTPTR, R0     | 0814                |      |  |
|  |  |  |      | 00        | 000E0    | BEQL   | 7\$                             |                     |      |  |
|  |  |  |      | 50        | 000E9    | MOVL   | R0, MODRSTPTR                   | 0817                |      |  |
|  |  |  | 6E   | 00000000G | 00       | SUBL3  | #1, DBG\$SRC_NEXT_LNUM, LINENO  | 0818                |      |  |
|  |  |  |      | 01        | C3       | BRW    | 18\$                            | 0785                |      |  |
|  |  |  |      | 00E0      | 31       | CLRB   | NAME_BUF                        | 0830                |      |  |
|  |  |  |      | 04        | AE       | MOVZWL | 4(R4), STRING_PTR               | 0831                |      |  |
|  |  |  |      | 04        | A4       | MOVZWL | (R4), LENGTH                    | 0832                |      |  |
|  |  |  |      | 57        | 64       | TSTL   | LENGTH                          | 0837                |      |  |
|  |  |  |      | 57        | D5       | BLEQ   | 10\$                            |                     |      |  |
|  |  |  |      | 57        | D5       | MOVZBL | (STRING_PTR)+, CHAR             | 0839                |      |  |
|  |  |  |      | 55        | 88       | DECL   | LENGTH                          | 0840                |      |  |
|  |  |  |      | 57        | 9A       | CMPL   | CHAR, #32                       | 0841                |      |  |
|  |  |  |      | 20        | 00105    | BEQL   | 9\$                             | 0849                |      |  |
|  |  |  |      | 57        | D7       | TSTL   | LENGTH                          |                     |      |  |
|  |  |  |      | 57        | 00108    | BNEQ   | 11\$                            |                     |      |  |
|  |  |  |      | 55        | D1       | PUSHAB | P_AAL                           | 0851                |      |  |
|  |  |  |      | 20        | 0010A    | PUSHL  | #1                              |                     |      |  |
|  |  |  |      | F2        | 13       | 0010D  | PUSHL                           | #164706             |      |  |
|  |  |  |      | 57        | D5       | 0010F  | CALLS                           | #3, LIB\$SIGNAL     |      |  |
|  |  |  |      | 12        | 12       | 00111  | TSTL                            | LENGTH              | 0857 |  |
|  |  |  |      | 04        | AA       | 00113  | BLEQ                            | 13\$                |      |  |
|  |  |  |      | 01        | DD       | 00116  | CMPL                            | CHAR, #92           | 0859 |  |
|  |  |  |      | 00028362  | 8F       | 00118  | BEQL                            | 13\$                |      |  |
|  |  |  |      | 00        | 0000000G | 03     | CALLS                           | #3, LIB\$SIGNAL     |      |  |
|  |  |  |      | 57        | FB       | 0011E  | TSTL                            | LENGTH              |      |  |
|  |  |  |      | 57        | D5       | 00125  | BLEQ                            | 13\$                |      |  |
|  |  |  |      | 41        | 15       | 00127  | CMPL                            | CHAR, #32           |      |  |
|  |  |  |      | 0000005C  | 8F       | 00129  | BEQL                            | 13\$                |      |  |
|  |  |  |      | 55        | D1       | 00130  | CMPL                            | CHAR, #32           |      |  |
|  |  |  |      | 20        | 38       | 00132  | BEQL                            | 13\$                |      |  |
|  |  |  |      | 55        | D1       | 00132  | INC8                            | NAME_BUF            | 0862 |  |
|  |  |  |      | 33        | 13       | 00135  | MOVZBL                          | NAME_BUF, R3        | 0863 |  |
|  |  |  |      | 50        | 53       | 00137  | CMPB                            | R3, #80             |      |  |
|  |  |  |      | 8F        | 04       | AE     | BLEQU                           | 12\$                |      |  |
|  |  |  |      | 53        | 91       | 0013A  | PUSHL                           | R4                  | 0865 |  |
|  |  |  |      | 54        | DD       | 00144  | CALLS                           | #1, DBG\$NNEXT_WORD |      |  |
|  |  |  |      | 00        | 0000000G | 01     | PUSHL                           | R0                  |      |  |
|  |  |  |      | 50        | DD       | 0014D  | PUSHL                           | #1                  |      |  |
|  |  |  |      | 00028E88  | 8F       | 00151  | PUSHL                           | #167608             |      |  |
|  |  |  |      | 00        | 0000000G | 03     | CALLS                           | #3, LIB\$SIGNAL     |      |  |
|  |  |  |      | 04        | AE       | 00157  | MOVB                            | CHAR, NAME_BUF[R3]  | 0866 |  |
|  |  |  |      | 55        | 90       | 0015E  | MOVZBL                          | (STRING_PTR)+, CHAR | 0867 |  |
|  |  |  |      | 55        | 88       | 00163  | DECL                            | LENGTH              | 0868 |  |
|  |  |  |      | 57        | D7       | 00166  |                                 |                     |      |  |

|           |           |                                                        |          |      |
|-----------|-----------|--------------------------------------------------------|----------|------|
| 0000005C  | 8F        | BB 11 00168                                            | BRB 11\$ | 0857 |
| 04        | A4        | 55 D1 0016A 13\$: CMPL CHAR, #92                       | 0874     |      |
| 64        |           | 2A 12 00171 BNEQ 14\$                                  |          |      |
|           |           | 58 D0 00173 MOVL STRING_PTR, 4(R4)                     | 0881     |      |
| 00000000G | 00        | 57 B0 00177 MOVW LENGTH, (R4)                          | 0882     |      |
| 56        |           | AE 9F 0017A PUSHAB NAME_BUF                            | 0887     |      |
|           |           | 01 FB 0017D CALLS #1, DBG\$STA GETSOURCEMOD            |          |      |
|           |           | 50 D0 00184 MOVL R0, MODRSTPTR                         |          |      |
|           |           | 38 12 00187 BNEQ 17\$                                  |          |      |
| 00000000G | 00        | AE 9F 00189 PUSHAB NAME_BUF                            | 0893     |      |
| 56        |           | 01 DD 0018C PUSHL #1                                   | 0895     |      |
|           |           | 000281E8 8F DD 0018E PUSHL #164328                     |          |      |
|           |           | 03 FB 00194 CALLS #3, LIB\$SIGNAL                      |          |      |
|           |           | 24 11 0019B BRB 17\$                                   |          |      |
| 52        | 00000000G | 00 DD 0019D 14\$: MOVL DBG\$GL_SCREEN_SOURCE, SRC_DISP | 0874     |      |
| 56        | 34        | 06 13 001A4 BEQL 15\$                                  | 0907     |      |
|           |           | A2 D0 001A6 MOVL 52(SRC_DISP), MODRSTPTR               | 0908     |      |
|           |           | 15 11 001AA BRB 17\$                                   | 0910     |      |
| 50        | 00000000G | 00 DD 001AC 15\$: MOVL DBG\$SRC_NEXT_MODRSTPTR, R0     | 0912     |      |
| 56        |           | 09 12 001B3 BNEQ 16\$                                  |          |      |
|           |           | 7E D4 001B5 CLRL -(SP)                                 | 0917     |      |
| 00000000G | 00        | 01 FB 001B7 CALLS #1, DBG\$STA GETSOURCEMOD            |          |      |
| 56        |           | 50 D0 001BE 16\$: MOVL R0, MODRSTPTR                   |          |      |
|           |           | 0C AC DD 001C1 17\$: PUSHL MESSAGE_VECT                | 0923     |      |
|           |           | 04 AE 9F 001C4 PUSHAB LINENO                           |          |      |
| 00000000G | 00        | 54 DD 001C7 PUSHL R4                                   |          |      |
| 04        |           | 03 FB 001C9 CALLS #3, DBG\$NSAVE_DECIMAL_INTEGER       |          |      |
| 50        |           | 50 E8 001D0 BLBS R0, 18\$                              |          |      |
|           |           | 04 D0 001D3 MOVL #4, R0                                | 0925     |      |
|           |           | 04 001D6 RET                                           |          |      |
| 04        | A9        | 56 D0 001D7 18\$: MOVL MODRSTPTR, 4(NOUN_NODE)         | 0932     |      |
| 69        |           | 6E D0 001DB MOVL LINENO, (NOUN_NODE)                   | 0933     |      |
| 50        |           | 01 D0 001DE MOVL #1, R0                                | 0938     |      |
|           |           | 04 001E1 RET                                           | 0939     |      |

: Routine Size: 482 bytes, Routine Base: DBG\$CODE + 0224

812 0940 1 GLOBAL ROUTINE DBGSNEXECUTE\_EDIT (VERB\_NODE, MESSAGE\_VECT) =  
813 0941 1  
814 0942 1 Functional Description  
815 0943 1  
816 0944 1 This routine performs the action associated with the EDIT  
817 0945 1 command. EDIT is the interface to the EDITH editor.  
818 0946 1  
819 0947 1 This routine constructs a DCL command line which invokes EDITH.  
820 0948 1 Either LIB\$SPAWN or LIB\$DO\_COMMAND is called to execute the  
821 0949 1 command line.  
822 0950 1  
823 0951 1 Formal Parameters  
824 0952 1  
825 0953 1 VERB\_NODE - A longword containing the address of the  
826 0954 1 verb node of the command execution tree.  
827 0955 1 This tree has already been built by the  
828 0956 1 DBGSNPARSE\_EDIT routine.  
829 0957 1  
830 0958 1 MESSAGE\_VECT - The address of a longword to contain the  
831 0959 1 address of an error message vector  
832 0960 1  
833 0961 1 Implicit Inputs  
834 0962 1  
835 0963 1 The command tree contains a verb node and a noun node.  
836 0964 1  
837 0965 1 Implicit Outputs  
838 0966 1  
839 0967 1 This routine constructs a command line and calls either  
840 0968 1 LIB\$SPAWN or LIB\$DO\_COMMAND to execute the command line.  
841 0969 1  
842 0970 1 Routine Value  
843 0971 1  
844 0972 1 A completion code.  
845 0973 1  
846 0974 1 Completion Codes  
847 0975 1  
848 0976 1 STSSK\_SUCCESS (1) - Success. Command executed  
849 0977 1 STSSK\_SEVERE (4) - Failure. The command could not be  
850 0978 1 executed. An error message is constructed.  
851 0979 1  
852 0980 1 Side Effects  
853 0981 1  
854 0982 1 If the EDIT/EXIT form of the command was given, the debugging  
855 0983 1 session is terminated.  
856 0984 1  
857 0985 1  
858 0986 2 BEGIN  
859 0987 2  
860 0988 2  
861 0989 2 MAP  
862 0990 2 VERB\_NODE : REF DBGSVERB\_NODE;  
863 0991 2  
864 0992 2 LOCAL  
865 0993 2 COMMAND\_LINE: VECTOR[132,BYTE], ! Command line to invoke EDITH  
866 0994 2 CTRLBUF,  
867 0995 2 CTRLDESC: DBGSSTG\_DESC,  
868 0996 2 FABPTR: REF SFAB DECL, ! FAB for searching for newer file ver.  
FAOBUF: VECTOR[80, BYTE],

```
869 0997 2 FAODESC: DBG$STG_DESC,
870 0998 2 FAOLEN,
871 0999 2 LENGTH,
872 1000 2 LINENO,
873 1001 2 MODRSTPTR,
874 1002 2 NAMESA,
875 1003 2 NAMPTR: REF $NAM_DECL,
876 1004 2 NAMRSA,
877 1005 2 NOUN_NODE : REF DBG$NOUN_NODE,
878 1006 2 OLDNAMPTR : REF $NAM_DECL,
879 1007 2 PTR,
880 1008 2 RECNO,
881 1009 2 STATUS,
882 1010 2 STG_DESC: DBG$STG_DESC;      ! String descriptor

883 1011 2
884 1012 2
885 1013 2
886 1014 2 ! Recover the noun node. From this, obtain the module RST pointer and
887 1015 2 line number.
888 1016 2 !
889 1017 2 NOUN_NODE = .VERB_NODE[DBG$L_VERB_OBJECT_PTR];
890 1018 2 MODRSTPTR = .NOUN_NODE[DBG$L_ADJECTIVE_PTR];
891 1019 2 LINENO = .NOUN_NODE[DBG$L_NOUN_VALUE];
892 1020 2
893 1021 2
894 1022 2 ! Build the command line and the string descriptor
895 1023 2 with which we will invoke EDITH. We start by initializing
896 1024 2 the current length of the command line and a pointer into
897 1025 2 the command line.
898 1026 2
899 1027 2 LENGTH = 0;
900 1028 2 PTR = COMMAND_LINE;
901 1029 2
902 1030 2
903 1031 2 ! Append the LSEDIT command to the line.
904 1032 2 Note - the command name may change. We leave a few extra
905 1033 2 spaces so we can patch in whatever command name they choose,
906 1034 2 up to 15 characters.
907 1035 2 Additional note - it now appears that the command name will
908 1036 2 be EDIT/LSE. However, this change will not happen until at least
909 1037 2 version 4.1 of VMS.
910 1038 2
911 1039 2 LENGTH = .LENGTH + 15;
912 1040 2 PTR = CHSMOVE(15, UPLIT BYTE(%ASCII 'LSEDIT      '), .PTR);
913 1041 2
914 1042 2
915 1043 2 ! The following block is only executed when a source file is present.
916 1044 2 Translate the module and line number into a NAM block pointer and record
917 1045 2 number within the file. This is done by calling a routine in DBGSOURCE.
918 1046 2 Also, do the checking for a more recent version of the source file,
919 1047 2 and issue any appropriate messages. After this is done, add the
920 1048 2 file name and the appropriate /START_POSITION qualifier.
921 1049 2
922 1050 2 IF .MODRSTPTR NEQ 0
923 1051 2 THEN
924 1052 3 BEGIN
925 1053 3   DBG$SRC_LINE_TO_REC(.MODRSTPTR, .LINENO, OLDNAMPTR, RECNO);
```

```
926 1054 3
927 1055 3
928 1056 3
929 1057 3
930 1058 3
931 1059 3
932 1060 3
933 1061 3
934 1062 3
935 1063 3
936 1064 3
937 1065 3
P 1066 3
P 1067 3
P 1068 3
P 1069 3
P 1070 3
P 1071 3
P 1072 3
P 1073 3
P 1074 3
P 1075 3
P 1076 3
949 1077 3
950 1078 3
951 1079 3
952 1080 3
953 1081 3
954 1082 3
955 1083 3
956 1084 3
957 1085 3
958 1086 3
959 1087 3
960 1088 3
961 1089 3
962 1090 3
963 1091 3
964 1092 3
965 1093 3
966 1094 3
967 1095 4
968 1096 4
969 1097 4
970 1098 4
971 1099 4
972 1100 4
973 1101 4
974 1102 4
975 1103 4
976 1104 4
977 1105 4
978 1106 4
979 1107 5
980 1108 5
981 1109 5
982 1110 4

| Get memory for the FAB and NAM blocks.
FABPTR = DBGSGET_TEMPMEM((FABSC-BLN+3)/4);
NAMPTR = DBGSGET_TEMPMEM((NAMSC-BLN+3)/4);
NAMESA = DBGSGET_TEMPMEM((NAMSC_MAXRSS+3)/4);
NAMRSA = DBGSGET_TEMPMEM((NAMSC_MAXRSS+3)/4);

| Initialize the FAB and NAM blocks for $PARSE and $SEARCH.
$FAB_INIT(FAB = .FABPTR,
           NAM = .NAMPTR,
           FNA = .OLDNAMPTR[NAMSL_ESA],
           FNS = (.OLDNAMPTR[NAM$B_ESL] - .OLDNAMPTR[NAM$B_VER]) );
$NAM_INIT(NAM = .NAMPTR,
           ESS = NAMSC_MAXRSS,
           ESA = .NAMESA,
           RSS = NAMSC_MAXRSS,
           RSA = .NAMRSA );

| Parse the spec, and then search for the most recent version.
| The PARSE should never fail, since we are using a string from a
| previous file OPEN. However, under very unusual circumstances,
| the SEARCH may fail (if a user spawned out from DEBUG and renamed
| or deleted the file, for example).
STATUS = $PARSE(FAB = .FABPTR);
IF NOT .STATUS
THEN
  SDBG_ERROR('DBGNTYPE\DBG$NEXECUTE_EDIT - RMS Parse error',.STATUS);
STATUS = $SEARCH(FAB = .FABPTR);
IF NOT .STATUS
THEN
  SIGNAL(DBGS_EDITNOFILE,2,
         .OLDNAMPTR[NAM$B_ESL],.OLDNAMPTR[NAMSL_ESA],.STATUS)
ELSE
  BEGIN
    | Issue a message about the file we are editing.
    SIGNAL(DBGS_EDITFILE,2,.NAMPTR[NAM$B_RSL],.NAMPTR[NAMSL_RSA]);
    | Now compare the result of the SEARCH with the old filename, and
    | issue a pair of messages if the version differs.
    IF CH$NEQ(.OLDNAMPTR[NAM$B_ESL], .OLDNAMPTR[NAMSL_ESA],
               .NAMPTR[NAM$B_RS], .NAMPTR[NAMSL_RS])
    THEN
      BEGIN
        SIGNAL(DBGS_EDITREVVER);
        SIGNAL(DBGS_EDITDISVER,2,.OLDNAMPTR[NAM$B_ESL],.OLDNAMPTR[NAMSL_ESA]);
      END;
```

```
983 1111 4
984 1112 4
985 1113 4
986 1114 4
987 1115 4
988 1116 4
989 1117 4
990 1118 4
991 1119 4
992 1120 4
993 1121 4
994 1122 4
995 1123 4
996 1124 4
997 1125 4
998 1126 4
999 1127 4
1000 1128 4
1001 1129 4
1002 1130 4
1003 1131 4
1004 1132 4
1005 1133 4
1006 1134 4
1007 1135 4
1008 1136 4
1009 1137 4
1010 1138 4
1011 1139 4
1012 1140 4
1013 1141 4
1014 1142 4
1015 1143 4
1016 1144 4
1017 1145 4
1018 1146 4
1019 1147 4
1020 1148 4
1021 1149 4
1022 1150 4
1023 1151 4
1024 1152 4
1025 1153 3
1026 1154 3
1027 1155 2
1028 1156 2
1029 1157 2
1030 1158 2
1031 1159 2
1032 1160 2
1033 1161 2
1034 1162 2
1035 1163 2
1036 1164 2
1037 1165 2
1038 1166 2
1039 1167 2

        ! Append the start position qualifier. Again, leave some
        ! extra space for flexibility.
        LENGTH = .LENGTH + 21;
        PTR = CH$MOVE(21, UPLIT BYTE(%ASCII '/START_POSITION = '), .PTR);

        ! We need to call $FAO
        ! to translate the record number that we have into a
        ! string.
        CTRLBUF = UPLIT BYTE(%ASCII '(!UL,1) ');
        CTRLDESC[DSC$B_CLASS] = DSC$K_CLASS_S;
        CTRLDESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
        CTRLDESC[DSC$W_LENGTH] = 8;
        CTRLDESC[DSC$A_POINTER] = .CTRLBUF;
        FAOLEN = 0;
        FAODESC[DSC$B_CLASS] = DSC$K_CLASS_S;
        FAODESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
        FAODESC[DSC$W_LENGTH] = 80;
        FAODESC[DSC$A_POINTER] = FAOBUF;
        IF NOT SYSSFAO (CTRLDESC, FAOLEN, FAODESC, .RECNO)
        THEN
            $DBG_ERROR('DBGNTYPE\DBG$NEXECUTE_EDIT $FAO failed');
        LENGTH = .LENGTH + .FAOLEN;
        IF .LENGTH GTR 132
        THEN
            $DBG_ERROR('EDIT command: Command line buffer has overflowed');
        PTR = CH$MOVE(.FAOLEN, FAOBUF, .PTR);

        ! Append the file name to the command line.
        LENGTH = .LENGTH + .NAMPTR[NAM$B_RSL];
        IF .LENGTH GTR 132
        THEN
            $DBG_ERROR('EDIT command: Command line buffer has overflowed');
        PTR = CH$MOVE(.NAMPTR[NAM$B_RSL], .NAMPTR[NAM$L_RSA], .PTR);
        END
        END

    ELSE
        SIGNAL(DBG$_EDITNOFILE);

        ! Now build a string descriptor with the command line.
        STG_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
        STG_DESC[DSC$A_DTYPE] = DSC$K_DTYPE_T;
        STG_DESC[DSC$W_LENGTH] = .LENGTH;
        STG_DESC[DSC$A_POINTER] = COMMAND_LINE;

        ! Developer bit 1 causes us to print the result but not
```

1040 1168 2 | do the action - (for testing purposes).  
1041 1169 2  
1042 1170 2 IF .DBG\$GL\_DEVELOPER[1]  
1043 1171 2 THEN  
1044 1172 3 BEGIN  
1045 1173 3 DBG\$PRINT(UPLIT BYTE(%ASCIC '!AS'), STG\_DESC);  
1046 1174 3 DBG\$NEWLINE();  
1047 1175 3 END  
1048 1176 3  
1049 1177 2 ELSE  
1050 1178 3 BEGIN  
1051 1179 3  
1052 1180 3 ! Set the screen back to normal  
1053 1181 3  
1054 1182 3 DBG\$SCR\_SCREEN\_NORMAL();  
1055 1183 3  
1056 1184 3 | Invoke EDITH with either LIB\$DO\_COMMAND or LIB\$SPAWN (depending  
1057 1185 3 on whether the /EXIT form of the command was used).  
1058 1186 3  
1059 1187 3 SELECTONE .VERB\_NODE[DBG\$B\_VERB\_COMPOSITE] OF  
1060 1188 3 SET  
1061 1189 3  
1062 1190 3 [EDIT\_EXIT]:  
1063 1191 4 BEGIN  
1064 1192 4 DBG\$GV\_CONTROL[DBG\$V\_CONTROL\_EXIT] = TRUE;  
1065 1193 4 LIB\$DO\_COMMAND(STG\_DESC);  
1066 1194 3 END:  
1067 1195 3  
1068 1196 3 [EDIT\_NOEXIT]:  
1069 1197 4 BEGIN  
1070 1198 4 LOCAL  
1071 1199 4 X:  
1072 1200 4 LIB\$SPAWN(STG\_DESC);  
1073 1201 4 IF .DBG\$GB\_KEYPAD\_INPUT  
1074 1202 4 THEN X = 1  
1075 1203 4 ELSE X = 0;  
1076 1204 4 SMG\$SET\_KEYPAD\_MODE(DBG\$GL\_KEYBOARD\_ID, X);  
1077 1205 3 END:  
1078 1206 3  
1079 1207 3 [OTHERWISE]:  
1080 1208 3 \$DBG\_ERROR('DBGNTYPE\DBG\$NEXECUTE\_EDIT');  
1081 1209 3 TES:  
1082 1210 2 END:  
1083 1211 2  
1084 1212 2  
1085 1213 2 | We are all done. Return success.  
1086 1214 2  
1087 1215 2 RETURN ST\$K\_SUCCESS  
1088 1216 1 END:

.PSECT DBG\$PLIT,NOWRT, SHR, PIC,0

20 20 20 20 20 20 20 20 20 20 54 49 44 45 53 40 00040 P.AAM: .ASCII \LSEDIT  
4E 24 47 42 44 5C 45 50 59 54 4E 47 42 44 2C 0004F P.AAN: .ASCII \,DBGNTYPE\<92>\DBG\$NEXECUTE\_EDIT - RMS \ :  
20 2D 20 54 49 44 45 5F 45 54 55 43 45 45 58 45 0005E

|                                                          |                                                           |
|----------------------------------------------------------|-----------------------------------------------------------|
| 4E 4F 49 54 72 6F 72 72 65 20 65 73 4D 52 0006D          | P.AAO: .ASCII \Parse_error\                               |
| 4E 24 47 42 44 50 59 54 45 5F 45 54 55 43 45 58 45 00071 | P.AAO: .ASCII \/START_POSITION = \                        |
| 4E 24 20 54 49 44 45 50 59 54 45 55 43 45 58 45 0007C    | P.AAP: .ASCII \(!UL,1) \                                  |
| 46 24 20 54 49 44 45 50 59 54 45 55 43 45 58 45 0008B    | P.AAQ: .ASCII \&DBGNTYPE\<92>\DBG\$NEXECUTE_EDIT \$FA0 f\ |
| 20 3A 64 6E 61 60 6D 6F 63 20 54 49 44 45 30 00091       | P.AAR: .ASCII \ailed\                                     |
| 75 62 20 65 6E 69 6C 20 64 6E 61 60 6D 6F 43 00099       | P.AAR: .ASCII \0EDIT command: Command line buffer has o\  |
| 20 3A 64 6E 61 60 6D 6F 63 20 54 49 44 45 30 000A8       | P.AAS: .ASCII \verflowed\                                 |
| 75 62 20 65 6E 69 6C 20 64 6E 61 60 6D 6F 43 00100       | P.AAS: .ASCII \0EDIT command: Command line buffer has o\  |
| 20 3A 64 6E 61 60 6D 6F 63 20 54 49 44 45 30 000E8       | P.AAT: .ASCII \verflowed\                                 |
| 75 62 20 65 6E 69 6C 20 64 6E 61 60 6D 6F 43 00119       | P.AAT: .ASCII <3>\!AS\                                    |
| 4E 24 47 42 44 50 59 54 4E 47 42 44 1A 00122             | P.AAU: .ASCII <26>\DBGNTYPE\<92>\DBG\$NEXECUTE_EDIT\      |
| 54 49 44 45 5F 45 54 55 43 45 58 45 00126                |                                                           |
| 4E 24 47 42 44 50 59 54 4E 47 42 44 1A 00126             |                                                           |
| 54 49 44 45 5F 45 54 55 43 45 58 45 00135                |                                                           |

.EXTRN SY\$PARSE, SY\$SEARCH

.PSECT DBG\$CODE,NOWRT, SHR, PIC,0

.ENTRY DBG\$NEXECUTE EDIT, Save R2,R3,R4,R5,R6,R7,- ; 0940

R8,R9,R10,R11

SP

1017

VERB NODE, R11

NOUN\_NODE

1018

4(NOUN\_NODE), MODRSTPTR

NOUN\_NODE, LINENO

1019

CLRL LENGTH

1027

COMMAND LINE, PTR

1028

#15, LENGTH

1039

ADDL2 #15, P.AAM, (PTR)

1040

MOVLC R3, PTR

MODRSTPTR

1050

TSTL BNEQ 1\$

BRW 8\$

1053

PUSHAB RECNO

OLDNAMPTR

MODRSTPTR, -(SP)

1058

CALLS #4 DBG\$SRC\_LINE\_TO\_REC

#20

PUSHL #1, DBG\$GET\_TEMP MEM

1059

CALLS #1, DBG\$GET\_TEMP MEM

R0, FABPTR

#24

MOVLC R0, NAMPTR

#64, -(SP)

1060

CALLS #1, DBG\$GET\_TEMP MEM

R0, NAMESA

#64, -(SP)

1061

CALLS #1, DBG\$GET\_TEMP MEM

R0, NAMRSA

#0, (SP), #0, #80, (FABPTR)

1069

0050 8F

00

6E

00 2C 00077

00

MOVCS

#0, (SP), #0, #80, (FABPTR)

1069

CALLS #1, DBG\$GET\_TEMP MEM

R0, NAMESA

#64, -(SP)

1061

CALLS #1, DBG\$GET\_TEMP MEM

R0, NAMRSA

#0, (SP), #0, #80, (FABPTR)

1069

|      |    |    |    |    |              |             |      |             |             |               |                                 |      |
|------|----|----|----|----|--------------|-------------|------|-------------|-------------|---------------|---------------------------------|------|
| 0060 | 8F | 34 | A8 | 00 | 16           | 68          | 5003 | 68          | 0007E       | MOVW          | #20483, (FABPTR)                | 1075 |
|      |    |    |    |    | 1F           | A8          |      | 02          | 0007F       | MOVB          | #2, 22(FABPTR)                  |      |
|      |    |    |    |    | 28           | A8          |      | 02          | 00084       | MOVB          | #2, 31(FABPTR)                  |      |
|      |    |    |    |    | 56           | 56          | 0C   | 57          | 00088       | MOVL          | NAMPTR, 40(FABPTR)              |      |
|      |    |    |    |    | 2C           | A8          | 0C   | AE          | 00090       | MOVL          | OLDNAMPTR, R6                   |      |
|      |    |    |    |    | 08           | A6          | 3D   | A6          | 00094       | MOVL          | 12(R6), 44(FABPTR)              |      |
|      |    |    |    |    | 6E           | 6E          |      | 00          | 00099       | SUBB3         | 61(R6), 11(R6), 52(FABPTR)      |      |
|      |    |    |    |    |              |             |      | 2C          | 000A0       | MOVCS         | #0, (SP), #0, #96, (NAMPTR)     |      |
|      |    |    |    |    |              |             |      | 67          | 000A7       |               |                                 |      |
|      |    |    |    |    | 67           |             | 6002 | 8F          | 000A8       | MOVW          | #24578, (NAMPTR)                |      |
|      |    |    |    |    | 02           | A7          |      | 01          | 000AD       | MNEG8         | #1, 2(NAMPTR)                   |      |
|      |    |    |    |    | 04           | A7          |      | 6E          | 000B1       | MOVL          | NAMRSA, 4(NAMPTR)               |      |
|      |    |    |    |    | 0A           | A7          |      | 01          | 000B5       | MNEG8         | #1, 10(NAMPTR)                  |      |
|      |    |    |    |    | 0C           | A7          | 04   | AE          | 000B9       | MOVL          | NAMESA, 12(NAMPTR)              |      |
|      |    |    |    |    |              |             |      | 58          | 000BE       | PUSHL         | FABPTR                          |      |
|      |    |    |    |    | 00000000G    | 00          |      | 01          | FB          | CALLS         | #1, SYSSPARSE                   |      |
|      |    |    |    |    | 52           |             |      | 50          | 000C0       | MOVL          | R0, STATUS                      |      |
|      |    |    |    |    | 17           |             |      | 52          | 000C7       | BLBS          | STATUS, 2\$                     |      |
|      |    |    |    |    |              |             |      | 52          | 000CA       | PUSHL         | STATUS                          |      |
|      |    |    |    |    |              |             |      | 52          | 000CD       | PUSHAB        | P.AAN                           |      |
|      |    |    |    |    |              |             |      | EF          | 000CF       | PUSHL         | #1                              |      |
|      |    |    |    |    |              |             |      | 01          | 000D5       | PUSHL         | #164706                         |      |
|      |    |    |    |    | 00000000G    | 00          |      | 04          | FB          | CALLS         | #4, LIB\$SIGNAL                 |      |
|      |    |    |    |    | 00000000G    | 00          |      | 58          | DD          | PUSHL         | FABPTR                          |      |
|      |    |    |    |    | 00000000G    | 00          |      | C1          | FB          | CALLS         | #1, SYSSSEARCH                  |      |
|      |    |    |    |    | 52           |             |      | 50          | 000E4       | MOVL          | R0, STATUS                      |      |
|      |    |    |    |    | 1B           |             |      | 52          | 000ED       | BLBS          | STATUS, 3\$                     |      |
|      |    |    |    |    |              |             |      | 52          | 000F0       | PUSHL         | STATUS                          |      |
|      |    |    |    |    |              |             |      | 52          | 000F3       | PUSHL         | 12(R6)                          |      |
|      |    |    |    |    |              |             |      | A6          | 000F5       | MOVZBL        | 11(R6), -(SP)                   |      |
|      |    |    |    |    | 7E           | 08          |      | A6          | 9A 000F8    | PUSHL         | #2                              |      |
|      |    |    |    |    |              |             |      | 02          | 000FC       | PUSHL         | #167947                         |      |
|      |    |    |    |    | 00000000G    | 00          |      | 05          | FB          | CALLS         | #5, LIB\$SIGNAL                 |      |
|      |    |    |    |    | 00000000G    | 00          |      | 0115        | 31 0010B    | BRW           | 9\$                             |      |
|      |    |    |    |    | 00000000G    | 00          |      | 04          | A7 DD 0010E | PUSHL         | 4(NAMPTR)                       |      |
|      |    |    |    |    | 00000000G    | 00          |      | 03          | A7 9A 00111 | MOVZBL        | 3(NAMPTR), -(SP)                |      |
|      |    |    |    |    | 00000000G    | 00          |      | 02          | DD 00115    | PUSHL         | #2                              |      |
|      |    |    |    |    | 00000000G    | 00          |      | 04          | FB 00117    | PUSHL         | #167915                         |      |
|      |    |    |    |    | 00000000G    | 00          |      | 51          | 0011D       | CALLS         | #4, LIB\$SIGNAL                 |      |
|      |    |    |    |    | 51           | 08          |      | A6 9A 00124 | MOVZBL      | 11(R6), R1    |                                 |      |
|      |    |    |    |    | 50           | 03          |      | A7 9A 00128 | MOVZBL      | 3(NAMPTR), R0 |                                 |      |
|      |    |    |    |    |              |             |      | 51          | 2D 0012C    | CMPCS         | R1, 012(R6), #0, R0, 04(NAMPTR) |      |
|      |    |    |    |    |              |             |      | 04          | B7 00132    | BEQL          | 4\$                             |      |
|      |    |    |    |    |              |             |      | 23          | 13 00134    | PUSHL         | #167923                         |      |
|      |    |    |    |    | 00000000G    | 00          |      | 01          | FB 00136    | CALLS         | #1, LIB\$SIGNAL                 |      |
|      |    |    |    |    | 00000000G    | 00          |      | 0C          | A6 DD 00143 | PUSHL         | 12(R6)                          |      |
|      |    |    |    |    | 7E           | 08          |      | A6 9A 00146 | MOVZBL      | 11(R6), -(SP) |                                 |      |
|      |    |    |    |    |              |             |      | 02          | DD 0014A    | PUSHL         | #2                              |      |
|      |    |    |    |    | 00000000G    | 00          |      | 04          | FB 0014C    | PUSHL         | #167931                         |      |
|      |    |    |    |    | 00000000G    | 00          |      | 59          | 15 C0 00159 | CALLS         | #4, LIB\$SIGNAL                 |      |
|      |    |    |    |    | 00000000G    | 00          |      | 15          | 28 0015C    | ADDL2         | #21, LENGTH                     |      |
|      |    |    |    |    | 6A 00000000' | EF          |      | 53          | 00164       | MOVCS         | #21, P.AAO, (PTR)               |      |
|      |    |    |    |    | 5A           |             |      | 50          | 00000000'   | MOVL          | R3, PTR                         |      |
|      |    |    |    |    | 0080         | CE 010E0008 |      | EF          | 9E 00167    | MOVAB         | P.AAP, CTRLBUF                  |      |
|      |    |    |    |    |              |             |      | 8F          | 00 0016E    | MOVL          | #17694728, CTRLDESC             |      |

|      |    |           |           |          |           |                     |          |                             |                          |      |
|------|----|-----------|-----------|----------|-----------|---------------------|----------|-----------------------------|--------------------------|------|
| 0084 | CE | 50        | D0        | 00177    | MOVL      | CTRLBUF, CTRLDESC+4 | 1128     |                             |                          |      |
| 24   | AE | 10        | AE        | D4 0017C | CLRL      | FAOLEN              | 1129     |                             |                          |      |
| 28   | AE | 010E0050  | 8F        | D0 0017F | MOVL      | #17694800, FAODESC  | 1132     |                             |                          |      |
|      |    | 30        | AE        | 9E 00187 | MOVAB     | FAOBUF, FAODESC+4   | 1133     |                             |                          |      |
|      |    | 08        | AE        | DD 0018C | PUSHL     | RECNO               | 1134     |                             |                          |      |
|      |    | 28        | AE        | 9F 0018F | PUSHAB    | FAODESC             |          |                             |                          |      |
|      |    | 18        | AE        | 9F 00192 | PUSHAB    | FAOLEN              |          |                             |                          |      |
|      |    | 00000000G | 00        | 008C     | CE        | 9F 00195            | PUSHAB   | CTRLDESC                    |                          |      |
|      |    | 15        |           | 04       | FB        | 00199               | CALLS    | #4, SYSSFAO                 |                          |      |
|      |    |           |           | 50       | E8        | 001A0               | BLBS     | R0, 5\$                     |                          |      |
|      |    |           |           | EF       | 9F        | 001A3               | PUSHAB   | P.AAO                       | 1136                     |      |
|      |    |           |           | 01       | DD        | 001A9               | PUSHL    | #1                          |                          |      |
|      |    | 00000000G | 00        | 00028362 | 8F        | DD 001AB            | PUSHL    | #164706                     |                          |      |
|      |    | 59        |           | 03       | FB        | 001B1               | CALLS    | #3, LIB\$SIGNAL             |                          |      |
|      |    | 00000084  | 8F        | 10       | AE        | C0 001B8            | ADDL2    | FAOLEN, LENGTH              | 1137                     |      |
|      |    |           |           | 59       | D1        | 001BC               | CMPL     | LENGTH, #132                | 1138                     |      |
|      |    |           |           | 15       | 15        | 001C3               | BLEQ     | 6\$                         |                          |      |
|      |    |           |           | EF       | 9F        | 001C5               | PUSHAB   | P.AAR                       | 1140                     |      |
|      |    |           |           | 01       | DD        | 001CB               | PUSHL    | #1                          |                          |      |
|      |    | 6A        | 00000000G | 00       | 00028362  | 8F                  | DD 001CD | PUSHL                       | #164706                  |      |
|      |    | 30        | AE        | 10       | AE        | 28 00 DA            | CALLS    | #3, LIB\$SIGNAL             |                          |      |
|      |    |           | 5A        |          | 53        | D0 001E0            | MOV3     | FAOLEN, FAOBUF, (PTR)       | 1141                     |      |
|      |    |           | 50        |          | 03        | A7 9A 001E3         | MOVL     | R3, PTR                     |                          |      |
|      |    |           | 59        |          | 50        | C0 001E7            | MOVZBL   | 3(NAMPTR), R0               | 1146                     |      |
|      |    | 00000084  | 8F        |          | 59        | D1 001EA            | ADDL2    | R0, LENGTH                  |                          |      |
|      |    |           |           |          | 15        | 15 001F1            | CMPL     | LENGTH, #132                | 1147                     |      |
|      |    |           |           |          | EF        | 9F 001F3            | BLEQ     | 7\$                         |                          |      |
|      |    |           |           |          | 01        | DD 001F9            | PUSHAB   | P.AAS                       | 1149                     |      |
|      |    |           |           |          | 00000000G | 00                  | 00028362 | PUSHL                       | #164706                  |      |
|      |    |           |           |          | 03        | FB 00201            | CALLS    | #3, LIB\$SIGNAL             |                          |      |
|      |    | 6A        | 04        | B7       | 03        | A7 9A 00208         | MOVZBL   | 3(NAMPTR), R0               | 1150                     |      |
|      |    |           | 5A        |          | 50        | 28 0020C            | MOV3     | RO, @4(NAMPTR), (PTR)       |                          |      |
|      |    |           |           |          | 53        | D0 00211            | MOVL     | R3, PTR                     |                          |      |
|      |    |           |           |          | 0D        | 11 00214            | BRB      | 9\$                         | 1089                     |      |
|      |    |           |           |          | 00000000G | 00                  | 0002900B | PUSHL                       | #167947                  | 1156 |
|      |    |           | 1A        | AE       | 010E      | 8F                  | DD 00216 | CALLS                       | #1, LIB\$SIGNAL          |      |
|      |    |           | 18        | AE       |           | 8F                  | 80 00223 | MOVW                        | #270, STG DESC+2         | 1162 |
|      |    |           | 1C        | AE       | FF7C      | 59                  | B0 00229 | MOVW                        | LENGTH, STG DESC         | 1163 |
|      |    | 19        | 00000000G | 00       |           | CD                  | 9E 0022D | MOVAB                       | COMMAND LINE, STG DESC+4 | 1164 |
|      |    |           |           |          | 01        | E1 00233            | BBC      | #1, DBG\$GL_DEVELOPER, 10\$ | 1170                     |      |
|      |    |           |           |          | 18        | AE 9F 0023B         | PUSHAB   | STG DESC                    | 1173                     |      |
|      |    |           |           |          | 00000000G | 00                  | 00000000 | PUSHAB                      | P.AAT                    |      |
|      |    |           |           |          | EF        | 9F 0023E            | CALLS    | #2, DBG\$PRINT              |                          |      |
|      |    |           |           |          | 02        | FB 00244            | CALLS    | #0, DBG\$NEWLINE            | 1174                     |      |
|      |    |           |           |          | 00        | FB 0024B            | BRB      | 15\$                        | 1170                     |      |
|      |    |           |           |          | 69        | 11 00252            | CALLS    | #0, DBG\$SCR_SCREEN_NORMAL  | 1182                     |      |
|      |    |           |           |          | 00        | FB 00254            | MOVZBL   | 1(R11), R0                  | 1187                     |      |
|      |    |           |           |          | 01        | AB 9A 0025B         | CMPB     | RO, #1                      | 1190                     |      |
|      |    |           |           |          | 50        | 91 0025F            | BNEQ     | 11\$                        |                          |      |
|      |    |           |           |          | 13        | 12 00262            | BISB2    | #16, DBG\$GV_CONTROL        | 1192                     |      |
|      |    |           |           |          | 10        | 88 00264            | PUSHAB   | STG_DESC                    | 1193                     |      |
|      |    |           |           |          | 18        | AE 9F 0026B         | CALLS    | #1, LIB\$DO_COMMAND         | 1187                     |      |
|      |    |           |           |          | 01        | FB 0026E            | BRB      | 15\$                        | 1196                     |      |
|      |    |           |           |          | 46        | 11 00275            | CMPB     | RO, #2                      |                          |      |
|      |    |           |           |          | 02        | 50 91 00277         | BNEQ     | 14\$                        |                          |      |
|      |    |           |           |          | 18        | AE 9F 0027C         | PUSHAB   | STG_DESC                    | 1200                     |      |

|           |    |           |           |       |       |                          |                     |                            |
|-----------|----|-----------|-----------|-------|-------|--------------------------|---------------------|----------------------------|
| 00000000G | 00 | 01        | FB        | 0027F | CALLS | #1, LIB\$SPAWN           |                     |                            |
| 14        | AE | 06        | 00000000G | 00    | E9    | 00286                    | BLBC                | DBG\$GB_KEYPAD_INPUT, 12\$ |
|           |    | 01        | DD        | 0028D | MOVL  | #1, X                    |                     |                            |
|           |    | 03        | 11        | 00291 | BRB   | 13\$                     |                     |                            |
|           |    | 14        | AE        | D4    | 00293 | 12\$:                    | CLRL                | X                          |
|           |    | 14        | AE        | 9F    | 00296 | 13\$:                    | PUSHAB              | X                          |
| 00000000G | 00 | 00000000G | 00        | 9F    | 00299 | PUSHAB                   | DBG\$GL_KEYBOARD_ID |                            |
| 00000000G | 00 | 02        | FB        | 0029F | CALLS | #2, SMG\$SET_KEYPAD_MODE |                     |                            |
|           |    | 15        | 11        | 002A6 | BRB   | 15\$                     |                     |                            |
| 00000000  | EF | 00000000  | EF        | 9F    | 002A8 | 14\$:                    | PUSHAB              | P.AAU                      |
|           | 01 | DD        | 002AE     | PUSHL | #1    |                          |                     |                            |
| 00000000G | 00 | 00028362  | 8F        | DD    | 002B0 | PUSHL                    | #164706             |                            |
|           | 50 | 03        | FB        | 002B6 | CALLS | #3, LIB\$SIGNAL          |                     |                            |
|           |    | 01        | DD        | 002BD | MOVL  | #1, R0                   |                     |                            |
|           |    | 04        | 002C0     | 15\$: | RET   |                          |                     |                            |

; Routine Size: 705 bytes, Routine Base: DBG\$CODE + 0406

; 1089 1217 0 END ELUDOM

.EXTRN LIB\$SIGNAL

## PSECT SUMMARY

| Name      | Bytes | Attributes                                               |
|-----------|-------|----------------------------------------------------------|
| DBG\$PLIT | 321   | NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(0)   |
| DBG\$OWN  | 81    | NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON, PIC,ALIGN(2) |
| DBG\$CODE | 1735  | NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(0)   |

## Library Statistics

| File                                     | ----- Symbols ----- |        |         | Pages<br>Mapped | Processing<br>Time |
|------------------------------------------|---------------------|--------|---------|-----------------|--------------------|
|                                          | Total               | Loaded | Percent |                 |                    |
| -\$255\$DUA28:[SYSLIB]LIB.L32:1          | 18619               | 68     | 0       | 1000            | 00:01.9            |
| -\$255\$DUA28:[DEBUG.OBJ]STRU[DEF].L32:1 | 32                  | 0      | 0       | 7               | 00:00.1            |
| -\$255\$DUA28:[DEBUG.OBJ]DBGLIB.L32:1    | 1545                | 84     | 5       | 97              | 00:02.1            |
| -\$255\$DUA28:[DEBUG.OBJ]DSTRECRDS.L32:1 | 418                 | 0      | 0       | 31              | 00:00.3            |
| -\$255\$DUA28:[DEBUG.OBJ]DBGMSG.L32:1    | 386                 | 10     | 2       | 22              | 00:00.3            |
| -\$255\$DUA28:[DEBUG.OBJ]DBGGEN.L32:1    | 150                 | 0      | 0       | 12              | 00:00.3            |

; Information: 1  
; Warnings: 0  
; Errors: 0

## COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:DBGNTYPE/OBJ=OBJ\$:DBGNTYPE MSRC\$:DBGNTYPE/UPDATE=(ENHS:DBGNTYPE)

: Size: 1735 code + 402 data bytes  
: Run Time: 00:34.6  
: Elapsed Time: 01:35.8  
: Lines/CPU Min: 2109  
: Lexemes/CPU-Min: 15119  
: Memory Used: 252 pages  
: Compilation Complete

0089 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

DBGTYPE  
LIS

DBGSHOW  
LIS

DBGPARSER  
LIS